



UNIVERSIDADE SALVADOR – UNIFACS
PROGRAMA DE PÓS-GRADUAÇÃO
MESTRADO ACADÊMICO EM SISTEMAS E COMPUTAÇÃO

EDUARDO MAGALHÃES SAMPAIO

DESENVOLVIMENTO DE INTERFACE INOVADORA 3D
PARA ACESSAR CONTEÚDO DIGITAL

Salvador
2010

EDUARDO MAGALHÃES SAMPAIO

**DESENVOLVIMENTO DE INTERFACE INOVADORA 3D
PARA ACESSAR CONTEÚDO DIGITAL**

Dissertação apresentada ao Curso de Mestrado Acadêmico em Sistemas e Computação, Universidade Salvador - UNIFACS, como requisito parcial para obtenção do grau de Mestre.

Orientador: Prof. Thomas de Araújo Buck

Salvador
2010

TERMO DE APROVAÇÃO

EDUARDO MAGALHÃES SAMPAIO

DESENVOLVIMENTO DE INTERFACE INOVADORA 3D
PARA ACESSAR CONTEÚDO DIGITAL

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre,
Universidade Salvador - UNIFACS, pela seguinte banca examinadora:

Thomas de Araújo Buck - Orientador
Doutor rer. nat. em Informática pela Universität Tübingen
Universidade Salvador - UNIFACS

José Maria Nazar David
Doutor em Engenharia de Sistemas e Computação pela COPPE/Universidade Federal do
Rio de Janeiro
Universidade Salvador - UNIFACS

Lynn Rosalina Gama Alves
Doutora em Educação e Comunicação pela Universidade Federal da Bahia - UFBA
Universidade do Estado da Bahia - UNEB

Salvador, 10 de setembro de 2010

AGRADECIMENTOS

O autor desta dissertação gostaria de agradecer a FAPESB - Fundação de Amparo a Pesquisa do Estado da Bahia - pelo apoio dado no desenvolvimento desta pesquisa.

RESUMO

Esta dissertação discute uma proposta de um sistema que tem como foco tridimensionalizar aulas através de uma sintaxe e de um sistema visual para ser usado como ambiente de apoio à aprendizagem, gerador de conteúdo e como ferramenta de auxílio à aprendizagem. O objetivo do software é mostrar que, além da possibilidade de se ler um material, pode-se criar um ambiente virtual e através deste transmitir o conhecimento do material de uma forma similar àquela feita em aulas presenciais. O programa, batizado de *Digital Content 3D Renderer (DC3DR)*, foi desenvolvido com o uso de tecnologias livres, incluindo os padrões *X3D* e *H-ANIM*, e a linguagem de programação Java com a finalidade de converter conteúdos em um ambiente 3D com um professor virtual para explicar o material convertido.

Palavras-chave: x3d. Ead. dc3dr. Realidade virtual. Aula. xj3d. XML. Java. h-anim. Avatar.

ABSTRACT

This thesis will discuss the proposal for a system that focuses on three dimensional lessons through a syntax and a visual system to be used as an environment for learning, content generator and aid learning tool. The purpose of this software is to demonstrate that, besides the possibility of reading material, you can create a virtual environment and through this pass on the information of the material in a way akin to the ones done in classrooms.

The program, dubbed *Digital Content 3D Renderer (DC3DR)* was created using free technologies, including standards *X3D* and *H-ANIM*, and the Java programming language in order to convert content into a 3D environment with a virtual teacher to explain the converted material.

Keywords: x3d; distance learning; dc3dr; virtual reality; class; xj3d; xml; java; h-anim; avatar

LISTA DE FIGURAS

Figura 1 – Ilustrando a finalidade do projeto	14
Figura 2 - Sensorama.....	18
Figura 3 – Óculos estéreo e luvas virtuais.....	19
Figura 4 – Um exemplo de HMD	21
Figura 5 – Ambiente virtual.....	22
Figura 6 – Exemplo de avatar.....	25
Figura 7 – Fluxo do software.....	29
Figura 8 - Lively	30
Figura 9 – Second Life	31
Figura 10 - ExitReality	32
Figura 11 – Web.Alive	33
Figura 12 – 3DXPLORER.....	34
Figura 13 - There	34
Figura 14 – Open Cobalt	35
Figura 15 - Alice.....	36
Figura 16 – Java3D.....	40
Figura 17 – Flash	40
Figura 18 – 3DMLW	41
Figura 19 – Unity.....	42
Figura 20 – Croquet.....	43
Figura 21 – O3D	44
Figura 22 – X3D	44
Figura 23 - Estrutura do <i>LoAI</i>	49
Figura 24 – Exemplo de um arquivo <i>Syntax</i>	51
Figura 25 – Fluxo final do projeto.....	52
Figura 26 - Diagrama de Sequência do Sistema.....	53
Figura 27 – Arquivo <i>ptable.jpg</i>	55
Figura 28 – Sistema em execução	55

LISTA DE QUADROS

Quadro 1 – Classificação de sistemas de RV	20
Quadro 2 – Meios usados para práticas de EAD	26
Quadro 3 - Softwares Analisados	37

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	15
1.2	JUSTIFICATIVA	15
1.3	ORGANIZAÇÃO DA DISSERTAÇÃO.....	16
2	ESTADO DA ARTE.....	17
2.1	REALIDADE VIRTUAL	17
2.2	AMBIENTE VIRTUAL	21
2.3	MODELAGEM 3D.....	23
2.4	AVATAR.....	24
2.5	UTILIZAÇÃO DE VOZ SINTÉTICA	25
2.6	EAD	26
2.6.1	Taxonomia EAD.....	26
2.7	EAD E REALIDADE VIRTUAL	27
3	SOFTWARES ANALISADOS.....	30
3.1	LIVELY	30
3.2	SECOND LIFE	31
3.3	EXITREALITY	32
3.4	WEB.ALIVE.....	33
3.5	3DXPLORER	33
3.6	THERE.....	34

3.7	OPEN COBALT	35
3.8	ALICE.....	35
3.9	DISCUSSÃO SOBRE OS SOFTWARES ANALISADOS.....	36
4	FERRAMENTAS DE DESENVOLVIMENTO GRÁFICAS ANALISADAS	39
4.1	JAVA3D	39
4.2	FLASH.....	40
4.3	3DMLW.....	41
4.4	UNITY	42
4.5	CROQUET.....	42
4.6	O3D.....	43
4.7	X3D.....	44
4.8	OUTRAS FERRAMENTAS	45
5	O SISTEMA	46
5.1	ARQUIVO DE ENTRADA DE DADOS	47
5.2	MÉTODO DE FALA DO AVATAR	47
5.3	H-ANIM.....	48
5.4	LINGUAGEM SYNTAX	49
5.5	DETALHES DO SISTEMA.....	51
5.6	EXEMPLO DE USO	54
6	CONCLUSÃO.....	57
6.1	TRABALHOS FUTUROS	58
	REFERÊNCIAS	59

1 INTRODUÇÃO

O ensino é, de uma forma técnica e sucinta, uma atividade triádica que envolve um componente que ensine, um componente que receba o ensinamento, e um terceiro componente, o conteúdo. Esta atividade pode ser realizada presencialmente ou virtualmente, sendo esta última, segundo Chaves (1999), descrita da seguinte maneira: "ênfatiza-se mais (ou apenas) a distância no espaço e se propõe que ela seja contornada através do uso de tecnologias de telecomunicação e de transmissão de dados, voz (sons) e imagens (incluindo dinâmicas, isto é, televisão ou vídeo)."

Essa transmissão de conhecimento é uma prática que ocorre desde as sociedades tribais (SOUSA, 2010). Neste período, as crianças aprendiam, e ainda aprendem, imitando os adultos. A partir daí, muito se evoluiu para chegar ao momento em que nos espaços escolares temos uma pessoa dedicada para tal exercício, que intitulamos professor.

De acordo com Rosa (2010), uma das definições de professor é uma pessoa que ensina algo a outrem. Esta ação, na maioria das vezes, ocorria em locais denominados de salas de aula. Neste ambiente real normalmente podemos encontrar um conjunto de alunos dispostos a aprender, um professor disponível para ensinar e um conjunto de ferramentas que auxiliem a tarefa do professor. Essas ferramentas muitas vezes caracterizam a forma de ensinar do professor. Todavia, muitas delas dependem da tecnologia para existir, que por sua vez, depende da necessidade de facilitar a transmissão do conhecimento do professor para os alunos.

Por ausência de tecnologias digitais e telemáticas em tempos passados, o ensino era feito apenas de forma presencial. Avançando um pouco no tempo, antes da chegada a Internet, estes professores presenciais que antes usavam livros didáticos e quadros em salas de aula, passaram a buscar outros meios de realizar suas profissões, seja através de visitas a locais, como zoológicos e museus, ou do uso de aparelhos como projetores e televisões, diversificando a forma de ensinar. Não o bastante, as técnicas continuaram a evoluir até a chegada da Internet, possibilitando uma nova forma de ensinar.

Focando na modalidade que tem o virtual como cenário, que neste texto é definido como Ensino à Distância (EAD), inicialmente os professores apenas disponibilizavam livros e

outros materiais de leituras na Internet e ao aluno lhe restava ler (SOUZA, 2005). Esta atividade monótona e desinteressante acabou logo recebendo novos elementos para deixá-la mais dinâmica e atrativa, como o uso de sons e imagens, destacados por Chaves (1999). Seria possível transformar o EAD, oferecendo novas maneiras de apresentar e gerar conteúdos e melhorando conseqüentemente o aprendizado do aluno?

A partir deste questionamento, somando-se ao fato de estarmos na era da computação ubíqua, com o objetivo de integrar de forma transparente a informática e as pessoas, fazendo com que os professores estejam constantemente em busca de soluções tecnológicas para auxiliar no processo de ensino, um trabalho de pesquisa foi realizado para construir um software capaz de apresentar conteúdo de uma forma tridimensional, num esforço de mostrar uma nova alternativa de disponibilização de materiais de estudo para esses alunos.

O programa, batizado de *Digital Content 3D Renderer* (DC3DR), foi concebido com a finalidade de, através de um conteúdo em um formato qualquer, como por exemplo, um texto no formato PDF, transformar este conteúdo em um ambiente virtual (figura 1), contendo um professor virtual que estará neste ambiente para tentar transmitir o conhecimento do material fornecido para o usuário do programa, fazendo alusão ao que advém em um curso presencial.

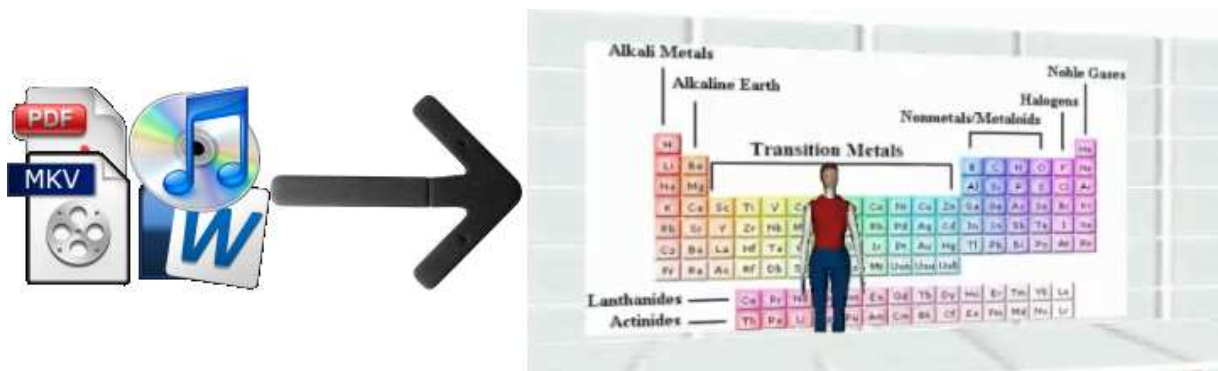


Figura 1 – Ilustrando a finalidade do projeto

1.1 OBJETIVOS

O objetivo geral deste trabalho foi fazer com que, através de um conteúdo qualquer, o computador seja capaz de transformar este material em uma aula virtual, com um professor virtual transmitindo o conhecimento do conteúdo. Para tal, o projeto se preocupa em desenvolver um aplicativo que, através de uma interface visual, consiga gerar um ambiente virtual, um humanóide virtual com a capacidade de falar, que fará o papel do professor, e uma forma de possibilitar a programação deste ambiente e deste humanóide. Para atingir o objetivo geral foram necessários os seguintes objetivos específicos:

- Selecionar uma ferramenta visual 3D para construção de um ambiente virtual.
- Escolher uma linguagem de programação.
- Adequar esta ferramenta visual à linguagem de programação.
- Desenvolver um ou mais locais virtuais, referenciando salas de aula.
- Elaborar um avatar, que terá a função de ser o professor.
- Criar uma linguagem intermediária que fará a ligação entre o conteúdo original e o ambiente virtual.
- Escolher uma ferramenta que possibilite transformar texto em áudio.

1.2 JUSTIFICATIVA

“Cada tecnologia modifica algumas dimensões da nossa inter-relação com o mundo, da percepção da realidade, da interação com o tempo e o espaço” (MORAN, 1995). Mas não basta somente a existência da tecnologia. Fatores econômicos e políticos influenciam esta evolução. Isto ocorreu, por exemplo, quando as ligações telefônicas ficaram mais acessíveis, fazendo com que o uso deste aparelho deixasse de ser algo usado em casos urgentes para passar a ser um meio de comunicação com pessoas distantes. Com o EAD não foi diferente. Agora que o uso de computadores e o acesso a Internet está difundido e tornou-se um requisito perante a sociedade, os obstáculos que a implementação do EAD enfrentava mudaram. É preciso disponibilizar novos meios para fazer com que o EAD consiga atrair estudantes e auxilie os professores a cumprir seus papéis.

Para justificar a concepção deste software, atualmente existe grande quantidade de conteúdo disponível em meio eletrônico. No entanto, há certa dificuldade em acessar essas informações, pois elas nem sempre se apresentam da forma mais apropriada para o leitor. Através de uma interface inovadora visual, este problema poderia ser amenizado.

É bastante comum que estudantes estabeleçam idéias erradas durante um aprendizado, pois existe uma dificuldade destes alunos de associar os conceitos apresentados durante os momentos de aprendizagem com os modelos concretos. Através dos recursos gráficos tridimensionais da Realidade Virtual (RV), este problema poderá ser amenizado.

Lima (1998), afirma que o termo RV foi criado pelo *Massachusetts Institute of Technology* (MIT) para propagar o conceito da presença humana em um ambiente gerado por computador. Queremos que o usuário do sistema perceba a alusão do ambiente com o que acontece nas salas de aula, através dessa RV.

Outro argumento seria a importância do EAD no Brasil, atualmente. De acordo com Mercado, Lira e Lira (2008), existem pelo menos 400 dissertações defendidas em programas de pós-graduação em educação voltadas para EAD. Isto mostra a preocupação, interesse e a necessidade de se obter resultados nesta área. Revela também que o investimento no EAD é importante e poderá resolver problemas na área de educação e problemas enfrentados por pessoas deficientes, permitindo que elas realizem tarefas que de outra forma não seriam possíveis.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

O capítulo 2 aborda a fundamentação teórica. Durante o capítulo 3, interfaces tridimensionais são analisadas, comparando suas implementações com as necessidades da interface proposta por este texto. O capítulo 4 analisa ferramentas para criação de interfaces gráficas, comparando-as entre si e, por fim, revelando a escolhida para ser usada na construção do *DC3DR*. O capítulo 5 discute a arquitetura e a linguagem usada no programa construído. E, por fim, o capítulo 6 conclui e fornece recomendações para trabalhos futuros além de sugerir como o *DC3DR* pode ser expandido e aprimorado.

2 ESTADO DA ARTE

Ao iniciar a pesquisa para a construção do sistema, algumas necessidades deveriam ser preenchidas. Um ambiente virtual foi imaginado com o objetivo de conceder ao usuário deste mesmo mundo virtual uma forma alternativa de absorção de conhecimento de forma lúdica. Neste ambiente, haveria um professor, que estaria ministrando a aula, assistida pelo usuário. O usuário estaria ali, virtualmente, naquele ambiente, assistindo a aula, da mesma maneira que o faz numa aula presencial. Para preencher estes requisitos o seguinte estudo foi feito.

2.1 REALIDADE VIRTUAL

RV pode ser definida como sendo a maneira mais avançada de interface do usuário de computador até agora disponível (HANCOCK, 1995). É uma forma que os usuários tem de visualizarem, manipularem e interagirem com computadores e dados (AUKSTAKALNIS; BLATNER, 1992). Segundo Jacobson, apud McLellan (2010), RV também pode ser definida como um ambiente gerado por um computador no qual o usuário sinta que faz parte deste ambiente. Com essa técnica, o usuário tem a habilidade de imergir, interagir e navegar no ambiente virtual gerado.

Estas três habilidades destacadas são importantes e distintas e devem ser bem definidas. A imersão significa que o usuário deve sentir-se dentro do ambiente, como se o ambiente fosse real. A interação permite o usuário manipular o ambiente e seus objetos, assim como o faz no mundo real. E a navegação habilita o usuário a deslocar-se pelo ambiente. Segundo Morie (1994), isoladamente, essas idéias não são exclusivas da RV, mas aqui elas coexistem.

A RV surgiu com os simuladores de vôo da United States Air Force (USAF). Um importante marco veio através do cineasta Morton Heilig, que, em 1960, querendo utilizar totalmente a visão do espectador, ao contrário dos 18% usados apenas visualizando um filme, criou um equipamento intitulado de Sensorama (figura 2).



Figura 2 - Sensorama

Fonte: <http://comm6480rpi.blogspot.com/2008/09/morton-heilig-pioneer-in-virtual.html>

Este equipamento fazia uso de três câmeras para ocupar todo o campo de visão do usuário, som estéreo, vibração, emitia cheiros e o usuário conduzia uma motocicleta fictícia, sentindo o vento bater no rosto através de um ventilador, provocando os sentidos sensoriais.

Apesar de ter feito pouco sucesso na época, foi um equipamento que provocou um investimento nesta área de RV. Em 1961, a Philco produziria um capacete com visor e, através de um sistema de circuito fechado de televisão, permitia a partir de movimentos com a cabeça do usuário, controlar uma câmera de televisão. Em 1968, Ivan Sutherland, prosseguindo a idéia de Heilig, construiu o primeiro capacete com imagens geradas por computador (SUTHERLAND, 2010).

Mas foi no final da década de 80 que Jaron Lanier (HALL, 1990) popularizou o termo RV, que, a partir deste momento, é usado para referenciar tipicamente mundos em três dimensões que fazem uso de óculos estéreo e luvas virtuais (figura 3).



Figura 3 – Óculos estéreo e luvas virtuais

Fonte: http://www.roboblock.co.kr/mall/m_mall_detail.php?ps_goid=1117

Através de Virtual Reality (2010), podemos classificar a RV na perspectiva de interface da seguinte maneira:

- Humano-máquina: ou sistemas monousuários, no qual a interface tenta estabelecer a comunicação entre o usuário e, por exemplo, o computador.
- Humano-(máquina)-humano: ou sistemas multiusuários, no qual a interface tem o objetivo de estabelecer uma comunicação entre dois ou mais usuários, e a máquina tem o papel de meio de comunicação entre eles.

Outra classificação proposta por Biocca e Levy (1995), relacionando a RV com a sua forma de exibição pode ser vista da forma apresentada no quadro 1:

Tipo	Descrição
Sistemas “janela”	O monitor do computador torna-se o meio para o mundo interativo e tridimensional. Computadores <i>desktop</i> são os mais usados e algumas vezes óculos 3D empregam efeitos estereoscópicos.
Sistemas “espelho”	O usuário se vê através da tela de projeção deslocando-se em um mundo virtual. Equipamentos de vídeo são usados para gravar os movimentos físicos do usuário.
Sistemas baseados em veículos	O usuário encontra-se em um veículo, como por exemplo, avião, carro, navio, e opera os controles deste veículo, simulando movimentos no mundo virtual.
Cavernas	O usuário fica dentro de uma sala cercada por telas grandes que projetam o mundo virtual de forma contínua.
Sistemas de RV totalmente imersivos	O usuário se equipa de dispositivos que geram estímulos através de computadores, como o estereoscópico <i>head-mounted display</i> (HMD), visto na figura 4.
Sistemas de realidade aumentada	O usuário equipa dispositivos visuais que adiciona objetos 3D virtuais em cenários reais.

Quadro 1 – Classificação de sistemas de RV

Fonte: Biocca e Levy (1995)

Segundo Christine Youngblut (1998), os sistemas “janela”, ou também chamados de *desktops* RV, em termos práticos, são mais adequados para uso em larga escala do que os sistemas de RV totalmente imersivos, pois o *hardware* usado pelos *desktops* RV são mais maduros e de fácil aquisição, enquanto que o *hardware* dos sistemas de RV totalmente imersivos são caros e ainda tentam chegar a um patamar de uso em larga escala. E a existência de mundos virtuais

na Internet já promove o uso destes *desktops* RV, tornando-o a opção ideal para se trabalhar com RV.

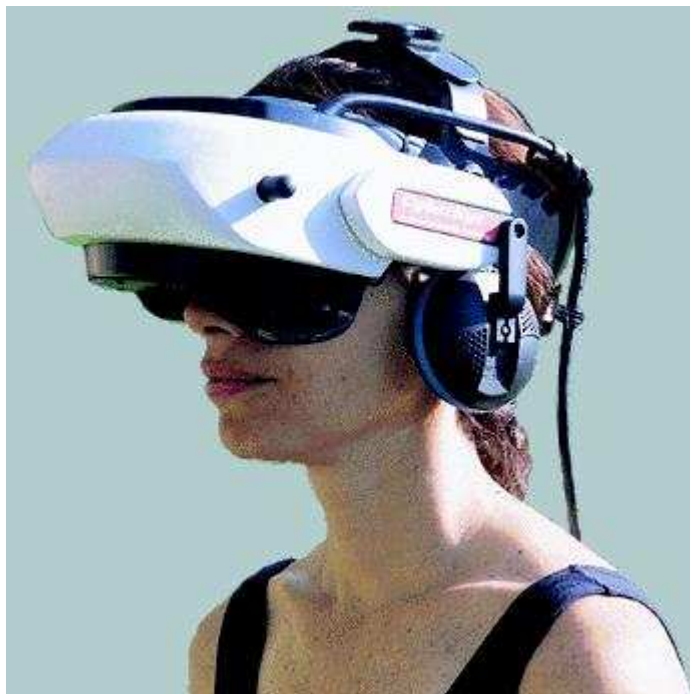


Figura 4 – Um exemplo de HMD
Fonte: <http://picsicio.us/domain/siggraph.org/>

2.2 AMBIENTE VIRTUAL

“Um ambiente virtual é um cenário gráfico, tridimensional e interativo gerado por computador. A sua exploração é feita através de um tipo de sistema de RV e constitui uma extensão das simulações convencionais numa tela de computador.” (TRINDADE; FIOLEAIS 2010).

Para uma percepção da realidade, o usuário do sistema deveria estar dentro de um ambiente virtual (figura 5). O emprego do ambiente virtual implica em uma experiência mais próxima do real que outros tipos de veiculação de informação, uma vez que o virtual "não se trata de modo algum de um mundo falso ou imaginário. Ao contrário, a virtualização é a dinâmica mesma do mundo comum, é aquilo através do qual compartilhamos uma realidade." (LÉVY, 1996).



Figura 5 – Ambiente virtual

Fonte: <http://www.metaversejournal.com/images/2010/03/springfling.jpg>

Esses ambientes em 3D possibilitam ao usuário um maior grau de ações dentro desta atmosfera. Quanto maior a capacidade de agir, maior a interatividade.

O ambiente virtual neste trabalho tem a função de representar o ambiente real de uma aula presencial, que normalmente é uma sala de aula. Mas esta aparência não é obrigatória. Pode ser que o ambiente virtual tenha que ser parecido com uma exposição de arte, caso a aula seja sobre pinturas, ou até mesmo com o espaço sideral, numa possível aula de astronomia, situação esta que similarmente só poderia ser alcançada através de uma visita a um planetário.

Para se aproximar um pouco mais da situação real, o usuário deveria poder navegar pelo cenário. Isto seria alcançado através da implementação de uma câmera virtual, fazendo com que o usuário possa se deslocar pelo ambiente e escolher o melhor ângulo para visualizar o mundo virtual a qualquer momento. Isto dá ao usuário uma liberdade e a possibilidade de analisar melhor o ambiente virtual, além de ser um item para aumentar a interatividade com o ambiente virtual.

Todavia, para criar um ambiente virtual, é necessário usar modelagens gráficas. Se possível, em três dimensões.

2.3 MODELAGEM 3D

A modelagem tridimensional é uma área da computação gráfica que tem como objetivo a geração de entidades em três dimensões, geração de cena estática e imagem em movimento com ou sem interatividade. Para elaboração são utilizadas ferramentas computacionais específicas para este tipo de tarefa. A modelagem é realizada através da criação de uma malha complexa de segmentos que dão forma ao objeto final.

Para chegar a tal nível de facilidade, a computação gráfica evoluiu bastante, desde a década de 60, quando ela foi criada. Mesmo nesta época ela demonstrava um imenso potencial, que era barrado devido ao alto custo do hardware e pouco conhecimento dos processos de geração de imagens, entre outros. E foi ainda nesta época que algoritmos como o de cálculo de visibilidade e a criação de grupos sediados em universidades americanas acontece. Na década seguinte temos Bézier desenvolvendo novas formas de representação de superfícies 3D generalizadas, Gouraud inventando um método de coloração de faces 3D e Catmull com seu método de visibilidade *z-buffer* e de geração de imagens intermediárias (*tweening*) (SAHIB, 2010). Até o final da década de 80, temos o filme *Tron*, que utiliza técnicas de computação gráfica e evolução do hardware gráfico, juntamente com novos algoritmos e a criação de pacotes gráficos e Application Programming Interface (APIs), como *OpenGL*, que descomplicam a programação gráfica. A partir da década de 90, a computação gráfica começa a ter participação não só no cinema, como também nas indústrias de vídeo, televisão e de entretenimento.

De volta ao presente, todavia, atualmente os programadores se utilizam de ferramentas gráficas, ou linguagens de modelagem para gerar as malhas complexas de segmentos. As ferramentas gráficas são utilizadas para criar o mundo virtual através do computador, descrevendo objetos existentes no mundo real. Desde que Ivan Sutherland percebeu que não era necessária a utilização de imagens analógicas, muito se investiu na área de computação gráfica, limitada pelo poder de processamento das máquinas antigas (SUTHERLAND, 2010). E para facilitar a exibição de objetos gráficos por programadores na tela, foram criadas linguagens de alto nível capazes de acessar o hardware gráfico. Com o uso dessas linguagens, fica transparente para o programador conceitos de computação gráfica e de hardware para a

geração de imagens na tela. O desenvolvedor agora se preocupa com o aspecto e o posicionamento dos objetos que ele deseja inserir no seu ambiente virtual.

A função da modelagem 3D neste trabalho é de inserir objetos 3D em todo o ambiente virtual. Na verdade, o ambiente virtual nada mais é do que a composição desses objetos. Para fazer que o usuário aparente estar vendo um ambiente real, o ambiente virtual deverá parecer como tal. No início existiam dificuldades de criação desses objetos, pois eles eram elaborados por pessoas da área de computação, que sabiam o que fazer para exibir os objetos na tela, mas não como construí-los. Com o surgimento de programas especializados em resolver esse tipo de situação, a necessidade de usar esses programadores desapareceu e aquilo que era apenas virtual começou a se assemelhar com o real. Todas essas etapas foram importantes para chegarmos ao patamar atual, no qual temos objetos virtuais que pouco se distinguem dos reais.

2.4 AVATAR

“Um avatar é um arquivo de imagem personalizado ou um processamento gráfico que representa um usuário. Os avatares são geralmente retratados de forma textual ou representações gráficas bidimensional ou tridimensional, embora outras formas sejam possíveis (sensações auditivas e táteis, por exemplo)” (FREE PATENTS ONLINE, 2010). “Avatar é uma pessoa artificial ou uma representação gráfica de uma pessoa real em um mundo virtual” (AVATAR, 2010).

Dentro do ambiente virtual proposto por este trabalho, utilizando técnicas de modelagem 3D, era uma necessidade ter um objeto que representasse o professor e os alunos. O avatar (figura 6) é o elemento que se encaixa nesta função, que é a de realizar gestos e se locomover dentro do ambiente virtual da mesma forma como uma pessoa se movimenta dentro da sala de aula e, o mais significativo, representar essa pessoa dentro do ambiente virtual.



Figura 6 – Exemplo de avatar

Fonte: http://www.frontiernet.net/~imaging/java3d_and_vrml.html

2.5 UTILIZAÇÃO DE VOZ SINTÉTICA

A utilização de voz sintética pode ser separada em análise e síntese de voz. A análise é o processo de converter a língua falada em um texto. Uma de suas etapas mais importantes é o processador de sinal, onde se faz a análise da frequência do som que o analisador recebe, para tentar reconhecer fonemas (MARANGONI, PRECIPITO, 2006).

A síntese de voz converte o texto escrito em uma língua falada. Também chamada de conversão *text-to-speech* (TTS), as principais etapas dessa ferramenta são a conversão do texto em um fonema e a geração do áudio.

A síntese de voz é uma técnica que permite o computador reproduzir a fala humana. É importante fazer uso desta tecnologia no sistema, pois os avatares precisam se comunicar. E o conteudista, que disponibilizará a aula, não necessitaria realizar a gravação da fala dos personagens envolvidos na aula. Apenas escreveria o texto e o computador transformaria aquele texto em áudio, com as respectivas vozes dos personagens. Ter o material de leitura

traduzido em som estabelece uma fronteira mais clara entre o documento que antes era lido, para um conteúdo agora multimídia, segundo Scavetta e Laufer (1997).

2.6 EAD

Depois de analisar o que há de mais moderno na parte digital, partimos para a parte de EAD.

Moore e Kearsley (1996) definem EAD da seguinte maneira:

[...] a família de métodos instrucionais onde as ações dos professores são executadas a parte das ações dos alunos, incluindo aquelas situações continuadas que podem ser feitas na presença dos estudantes. Porém, a comunicação entre o professor e o aluno deve ser facilitada por meios impressos, eletrônicos, mecânicos ou outros.

Esses métodos instrucionais surgiram no final do século XVII, entre as experiências de educação por correspondência. A partir do século XIX, houve um grande desenvolvimento nesta área, já sendo possível observar o uso de várias mídias, variando entre materiais impressos à mundos virtuais com a comunicação direta entre alunos e professores. Um importante marco do EAD no Brasil foi a regulamentação do mesmo através da Lei de Diretrizes e Bases da Educação em 1996. No quadro 2, podemos observar os diversos meios usados para praticar EAD em ordem cronológica.

Período	Recurso utilizado
1728	Correspondência
1929	Transmissão via rádio e fitas de áudio
1963	Transmissão via televisão e fitas de vídeo
1992	Internet

Quadro 2 – Meios usados para práticas de EAD

2.6.1 Taxonomia EAD

O EAD pode ser classificado de muitas formas. Todavia, apenas duas delas possuem uma relação direta com o software.

A primeira que iremos destacar é com relação ao sincronismo. Para que o EAD ocorra, podemos ter todos os participantes presentes e sincronizados através de alguma tecnologia, como videoconferência ou telefone, ou podemos ter um material à disposição do estudante para que ele faça a consulta no momento que melhor lhe convier. Este segundo seria o método assíncrono, pois não possui nenhuma dependência dos outros participantes. O programa é voltado para o método assíncrono, deixando livre o usuário para poder utilizar o conteúdo sem depender de outros usuários.

A segunda maneira destacada está relacionada aos meios de distribuição. De acordo com Martins e Moço (2009) no início do século XX, aqui no Brasil, as experiências com o EAD eram realizadas através de cartas, rádio e, mais tarde, pela TV para cursos profissionalizantes. Só com a chegada da internet banda larga é que se tornou viável a inclusão de cursos como graduações, sendo que mais de 93% deles utilizam a internet como o principal meio de ensino, dados estes destacados em Martins e Moço (2009). Através desses dados, justificamos a internet como meio de distribuição, além de que, atualmente, já é um meio de fácil acesso e globalmente utilizado para realização de pesquisas e estudos.

Do mesmo modo, a internet, como meio de distribuição, possibilita diversas formas para exibir o conteúdo. Dentre muitas, podemos listar e-mails, chats, fóruns e ambientes virtuais. Entre os listados, os ambientes virtuais são os que apresentam maior possibilidade de interação e a capacidade de aplicar todas as necessidades destacadas durante este capítulo.

2.7 EAD E REALIDADE VIRTUAL

Segundo Braga (2001), existem diversas razões para se usar a RV na educação, dentre as quais destaca:

- Maior motivação dos estudantes (usuários);
- O poder de ilustração da RV para alguns processos e objetos é muito maior do que outras mídias;
- Permite uma análise de muito perto;
- Permite uma análise de muito longe;

- Permite que as pessoas deficientes realizem tarefas que de outra forma não são possíveis;
- Dá oportunidades para experiências;
- Permite que o aprendiz desenvolva o trabalho no seu próprio ritmo;
- Não restringe o prosseguimento de experiências ao período da aula regular;
- Permite que haja interação, e desta forma estimula a participação ativa do estudante.

A RV é a ferramenta ideal para suprir múltiplas situações e contextos de pesquisa e aprendizagem. Através dela, não importa o estilo de aprendizagem do usuário. Ele pode ser verbal ou visual, de pesquisar ou de deduzir. A RV é capaz de fornecer todos eles para que o aluno escolha o seu método e possa aprender de uma maneira melhor.

De acordo com Christine Youngblut (1998), muitos pesquisadores e educadores acreditam que RV oferece grandes benefícios para a educação. Alguns deles mostram-se interessados na capacidade que a RV tem de facilitar as atividades do ensino construtivista. Outros acreditam na ampliação do ensino, alcançando alunos que aprendem melhor através de visualizações.

Nos ambientes tradicionais de estudo, os estudantes devem aprender através de assimilação, como, por exemplo, dentro de uma sala de aula escutando a transmissão de conhecimento do professor, ou através da leitura da bibliografia passada pelo mesmo. Os educadores, de uma maneira geral, acreditam que os estudantes são capazes de aprender e generalizar o novo conhecimento de uma forma melhor quando eles estiverem envolvidos pela situação.

Depois de toda essa discussão, o potencial da RV na educação foi estabelecido. No entanto, para gerar todos esses requisitos para que o aluno consiga obter todas as vantagens da RV é uma tarefa complicada, atualmente. Provavelmente seria mais fácil para os educadores continuarem a preparar os seus conteúdos das aulas da forma como eles já sabem fazer e usufruir de um software que capturasse esses conteúdos e gerasse automaticamente o mundo virtual, com tudo o que o avatar do professor teria que falar e gesticular durante a aula. O fluxo desta idéia pode ser observado na figura 7.

A figura 7 é muito parecida com a figura 1, vista no início do texto. Como mencionado anteriormente, esta é uma tarefa complicada. E para facilitar o seu desenvolvimento, um novo elemento foi introduzido no fluxo do software. O conteúdo agora será a entrada da criação de

3 SOFTWARES ANALISADOS

Antes da implementação do software, foi realizada uma pesquisa entre softwares de ambientes virtuais disponíveis. Mesmo que o propósito desses aplicativos não fosse compatível com o programa, o objetivo era poder utilizar a estrutura da parte gráfica e, através da programação, adequar todo este conjunto para criar o sistema. Para tal o software deveria possuir suporte a áudio, executar tanto em desktop como web, suportar algum tipo de conexão para que fosse possível trocar dados externamente, de custo gratuito, opensource para possibilitar modificações no código caso necessário, multi-plataforma para poder executar na maioria dos sistemas disponíveis e que pudesse ser programável com baixa complexidade. A seguir, vamos discutir os softwares pesquisados.

3.1 LIVELY



Figura 8 - Lively

Fonte: <http://www.nickburcher.com/2008/12/google-lively-rip-last-day-for-googles.html>

Lively (GOOGLE, 2009a) é um ambiente virtual no qual o usuário era capaz de controlar um avatar, manipular objetos predefinidos e conversar com outros avatares, através de textos, que estivessem no mesmo ambiente. A ideia do *Lively*, visível na figura 8, é agrupar até vinte pessoas dentro de uma sala, na qual o criador desta sala poderia utilizar objetos disponíveis no catálogo do programa, organizá-los e posicioná-los. Os outros usuários poderiam contribuir também com essa organização, fazendo com que esse ambiente mostrasse a personalidade desses usuários.

Suas limitações, como a falta de um ambiente programável, executar apenas no *Windows* e a impossibilidade de injetar objetos gerados por usuários, além de ter sido descontinuado, fez com que esta ferramenta não pudesse ser usada no projeto.

3.2 SECOND LIFE



Figura 9 – Second Life

Fonte: <http://www.mundotecno.info/noticias/o-second-life-esta-morto>

Second Life (SECOND Life, 2008) é um mundo virtual no qual o usuário controla um avatar que pode interagir com outros avatares. Ele pode ser classificado como um jogo, rede social, simulador, ambiente de ensino à distância e até como comércio virtual. No *Second Life*, figura 9, o usuário é capaz de gerar objetos virtuais e de lhes dar funcionalidades através do uso das ferramentas que o programa lhe oferece, e conversar via voz ou texto.

Second Life oferece muitas possibilidades para fazer com que o ensino à distância possa existir. Os avatares são muito realistas, a infra-estrutura já está pronta e é totalmente escalonável, capaz de receber uma enorme quantidade de usuários e a interface é intuitiva.

Apesar de toda essa estrutura, o *Second Life* infelizmente não oferece uma maneira de programar o ambiente. Especificamente para o nosso projeto, o *Second Life* deveria receber uma entrada, que seria a aula do professor e virtualizá-la. Mas a não existência de uma API de programação neste ambiente até o presente momento acaba eliminando qualquer possibilidade de que ele possa ser usado como interface para o projeto.

3.3 EXITREALITY

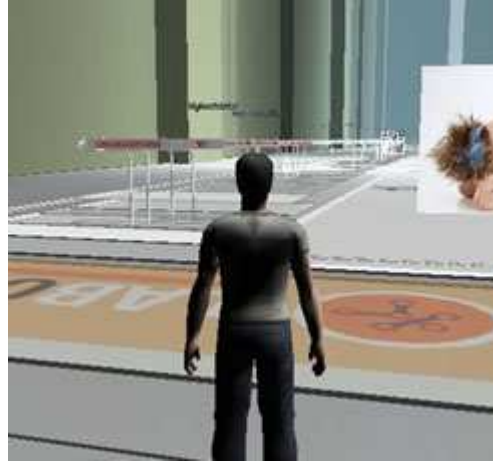


Figura 10 - ExitReality

Fonte: <http://counternotions.wordpress.com/2008/09/19/exitreality/>

ExitReality (EXIT Reality 2008) é um software que prega a criação de mundos 3D instantâneos de qualquer site. Cada usuário possui o seu espaço, que poderá configurar como bem desejar, principalmente se utilizando de outros objetos gerados por qualquer site virtualizado pela ferramenta.

Pelo que foi possível avaliar o sistema ExitReality, visível na Figura 10, depois de instalado precisa receber como parâmetro um site. Esse site será avaliado pelo servidor do *ExitReality* e algumas *tags* do *HTML* (HyperText Markup Language), como `<a>` e `` são transformadas em objetos tridimensionais. O usuário controla o seu avatar dentro deste ambiente e pode conversar via texto simultaneamente com outros usuários através de uma janela que abre juntamente com o ambiente. Esta janela de texto acaba tirando um pouco da imersão virtual normalmente proposta por esse tipo de ambiente.

O diferencial desta interface é que ela é capaz de receber um arquivo, neste caso HTML, do usuário, que seria o site a ser virtualizado, e criar um ambiente com base neste arquivo. Porém, o modo como o arquivo é interpretado não é programável e é feito através do servidor, excluindo a possibilidade de uso desta ferramenta no projeto.

3.4 WEB.ALIVE



Figura 11 – Web.Alive

Fonte: http://www.daden.co.uk/pages_virtual_worlds/virtual_worlds_gallery.html

Web.Alive (WEB.Alive 2008), visto na figura 11, é um aplicativo de mundos virtuais principalmente usado em e-commerce, aprendizado virtual e como ambiente colaborativo. Seu alto grau de imersão se deve ao fato de utilizar uma *engine* 3D aprimorada e som de alta qualidade.

No entanto, esta interface é proprietária e produzida apenas pela própria empresa. Conseqüentemente, ela é paga e deve ser encomendada para uma situação específica. Isto faz com que ela seja descartada para ser usada no projeto.

3.5 3DXPLORER

3Dxplorer (3DXPLORER 2008) (Figura 12) é uma ferramenta para criação de mundos virtuais. Utiliza Java e é capaz de receber conteúdo criado por usuários através de um kit gratuitamente fornecido. Também é possível usar programas de modelagem, como o *Google Sketchup* ou o *Autodesk 3D Studio* para criar seu próprio ambiente e importá-lo para o *3Dxplorer*.



Figura 12 – 3DXPLORER

Fonte: http://news.hiperia3d.com/2008_08_01_archive.html

Talvez esta seja a ferramenta que mais se aproxima dos requerimentos para ser utilizada no projeto. Mas o fato de que a informação deve ficar armazenada no servidor dos criadores deste software e a existência de um sistema de login também proprietário faz com que essa ferramenta seja de difícil implementação em qualquer site interno.

3.6 THERE



Figura 13 - There

Fonte: <http://www.there.com/info/homepage>

Visto como concorrente do *Second Life*, *There* (THERE, 2008) é uma interface para acessar um mundo virtual. Neste ambiente, ressaltado na figura 13, o usuário é representado por um avatar, que pode se deslocar, realizar gestos e modificar sua aparência. Os outros usuários conectados no mundo também são representados por avatares e podem interagir entre si através de textos e visualmente através dos movimentos dos seus avatares.

É uma interface proprietária e limitada quanto a uma personalização do ambiente através de programação. Por esta razão foi descartada como ferramenta para utilização no projeto.

3.7 OPEN COBALT



Figura 14 – Open Cobalt

Fonte: <http://web.duke.edu/~julian/Cobalt/Home.html>

Open Cobalt (Duke University, 2009) é um *browser* e um kit de construção para mundos virtuais (figura 14). É um sistema *open source* e gratuito, possui suporte a *VoIP* (Voice over IP), é multi-plataforma, assim como o Java e não necessita de um servidor, pois utiliza o protocolo P2P (Peer To Peer) para fazer a comunicação entre clientes.

A capacidade de criar mundos virtuais que este software tem é bastante ampla, mesmo na fase pouco madura de desenvolvimento na qual se encontra. Mas, pelo fato de ainda se encontrar neste estágio, já foi um motivo considerado para que ele não fosse usado no projeto. Outro detalhe é que, por ser amplo e focado mais em um ambiente colaborativo, sua complexidade na programação aumenta consideravelmente, fugindo do escopo inicial do projeto. Acabamos fazendo uma análise e percebemos que, ao tentar adequar esta ferramenta para resolver nossos problemas, seria tão ou mais trabalhoso quanto fazer outra. Por isso, ela foi descartada.

3.8 ALICE

Alice (ALICE.org, 2009) é um ambiente gratuito de programação 3D que tem como objetivo demonstrar o conceito de programação orientada a objeto. Criado pela Universidade Carnegie

Mellon, possui código aberto e seu desenvolvimento foi inspirado nas dificuldades iniciais dos estudantes que estão aprendendo a programar.

O que há de mais interessante é a interface de *Alice* (figura 15). Possui uma janela mostrando a cena com os objetos inseridos em tempo real. As funcionalidades são botões descritivos que são arrastados para um painel que administra todas as ações do sistema. Os objetos já possuem animações internas disponíveis para o usuário.



Figura 15 - Alice

Fonte: <http://www.developer.com/img/2007/04/java1516a01.jpg>

Entretanto, devido ao foco no aprendizado voltado para alunos de computação, esse sistema não se estabelece como ferramenta capaz de ser utilizada para realizar o propósito definido neste texto. Todavia, a interface é muito interessante e poderia servir de base para a construção da interface do programa.

3.9 DISCUSSÃO SOBRE OS SOFTWARES ANALISADOS

O Quadro 3 resume em alguns quesitos a capacidade de cada software analisado. *Lively*, *Second Life* e *There* são softwares que não possibilitam a inserção de informações por meio externo para poder alterar como o mundo virtual é. A interface visual é avançada e cada um possui uma boa base de usuários já cadastrados, mas existe certa dificuldade em alcançar o objetivo do software proposto por causa das restrições.

O *Exit Reality* tem uma vantagem, pois ele gera o mundo virtual através da página que você está visualizando. Deste modo, seria possível utilizar um arquivo no formato *HTML* para

poder manipular o mundo virtual criado pelo software. No entanto, a especificação *HTML* é muito limitada, e a forma de interpretação feita pelo *Exit Reality* fica responsável por um servidor o qual não se pode ter acesso para a modificação desses parâmetros.

O software *Web.Alive* funciona sob encomenda. Então, provavelmente, a parte de interpretação de conteúdo teria que ser construída para que alcançasse o objetivo do projeto. Contudo, por ser pago, não tivemos a chance de avaliá-lo melhor.

Atributos / Tecnologia	Audio	Ambiente	Conexão	Custo	Código aberto	Multi-plataforma	Programável
Lively	Sim	Desktop	Cliente-Servidor	Gratuito	Não	Não	Não
Second Life	Sim	Desktop	Cliente-Servidor	Gratuito	Não	Sim	Não
Exit Reality	Sim	Web	Cliente-Servidor	Gratuito	Não	Não	Não
Web.Alive	Sim	Web	Cliente-Servidor	A combinar	Não	N/A	Não
3Dxplorer	Sim	Web	Cliente-Servidor	Gratuito	Não	Sim	Não
There	Sim	Desktop	Cliente-Servidor	Gratuito	Não	Não	Não
Open Cobalt	Sim	Web/Desktop	P2P	Gratuito	Sim	Sim	Sim
Alice	Sim	Desktop	N/A	Gratuito	Sim	Sim	Sim

Quadro 3 - Softwares Analisados

3Dxplorer seria a ferramenta ideal, pois consegue receber objetos e parâmetros externamente. Mas o fato de possuir um sistema de login interno dificulta sua implementação em outros locais.

Open Cobalt e *Alice* são software gratuitos com grandes comunidades apoiando seus desenvolvimentos. Mas seus objetivos são, de certo modo, específicos. O *Open Cobalt* tem a função de criar mundos virtuais. Suas ferramentas são adequadas para tal, mas para conseguir implementar um método de leitura de arquivo e transformar este arquivo no mundo virtual mostrou-se um processo de certa complexidade, além do fato de que ele está numa fase inicial de desenvolvimento. Já o *Alice* possui objetos no seu formato proprietário e seu objetivo é

fazer com que o usuário monte o seu ambiente virtual e aprenda a usar os objetos. É um paradigma diferente do software proposto e, por este motivo, foi descartado.

Apesar de nenhum desses softwares analisados terem sido aproveitados, essa pesquisa foi necessária para estabelecer melhor o objetivo do software proposto e que sua construção foi uma necessidade, já que nenhum dos outros conseguiu suprir os requerimentos deste trabalho.

Vale a pena destacar que essa análise foi feita antes da construção do projeto. E, durante esse tempo, novos softwares foram surgindo. Dentre eles, mencionamos um artigo recentemente publicado (EHRLICH; MILLER, 2009) sobre um sistema virtual que ajuda no aprendizado de habilidades sociais. Apesar de o foco ser diferente do nosso sistema, sua estrutura e objetivos são muito parecidos com o sistema proposto neste texto.

4 FERRAMENTAS DE DESENVOLVIMENTO GRÁFICAS ANALISADAS

Neste capítulo serão analisadas algumas ferramentas de desenvolvimento gráficas que tenham a capacidade de serem usadas na construção do programa. A finalidade dessas ferramentas é de facilitar a criação do software, provendo operações básicas, como a exibição de objetos na tela, acesso a leitura de arquivos e operações de áudio. Como visto no capítulo 3, nenhum dos softwares conseguiu suprir as necessidades do programa proposto, fazendo com que seja necessário programá-lo por completo. Como o programa demanda requisitos visuais como condição principal para exibir o ambiente e os avatares, a busca por uma ferramenta para a construção do software deve conter uma capacidade gráfica, isto é, que seja especificamente voltada a exibição de imagens na tela.

Como a quantidade dessas ferramentas existentes é grande, a busca foi realizada com ferramentas bastante conhecidas e, de preferência, que fossem padrões já estabelecidos e de fácil adequação entre linguagens de programação.

4.1 JAVA3D

Java3D (JAVA, 2009) é uma API gráfica da linguagem Java. A vantagem de se usar esta API bastante conhecida é que ela já está integrada com uma linguagem multi-plataforma. O Java3D (figura 16) apenas se encarregaria de exibir o conteúdo tridimensional. Através da linguagem Java, a interpretação do conteúdo poderia ser alcançada.

Foi uma tecnologia considerada no projeto, todavia, o uso de padrões neste projeto era uma necessidade.

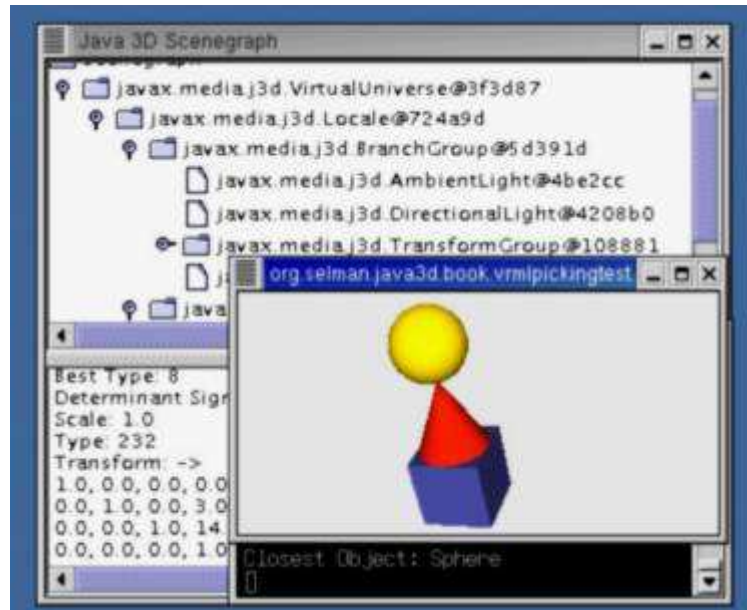


Figura 16 – Java3D

Fonte: http://www.frontiernet.net/~imaging/games_with_java3d.html

4.2 FLASH

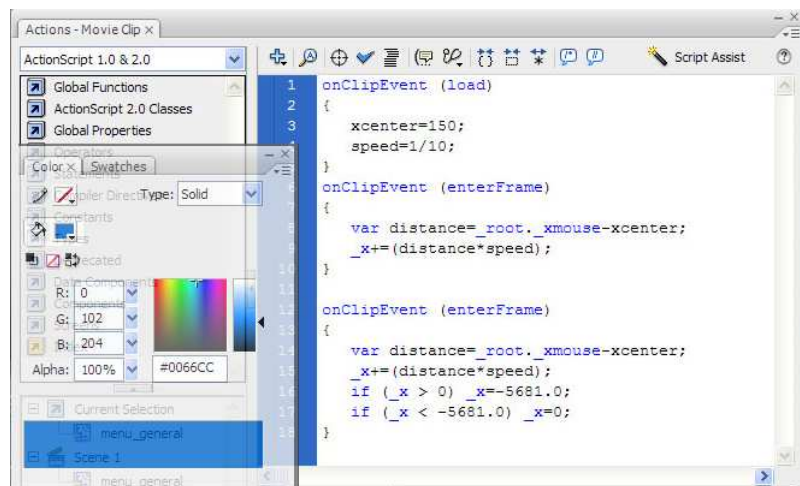


Figura 17 – Flash

Fonte: <http://www.sapphiresteel.com/Amethyst-Product-Page>

Flash (Adobe, 2009) (figura 17) é uma tecnologia que está no mercado há bastante tempo, tentando tornar-se um padrão. É um ambiente completo para construção visual e possui sua própria linguagem para gerar conteúdo dinâmico e seu player se encarrega da exibição.

Contudo, *Flash*, até a sua presente versão, continua sendo uma ferramenta de vetorização com algumas técnicas para dar uma sensação tridimensional. É paga e seu uso é pouco adequado para ser utilizado no projeto, já que não oferece diretamente suporte a ferramentas 3D, o que dificultaria o desenvolvimento do mesmo.

4.3 3DMLW

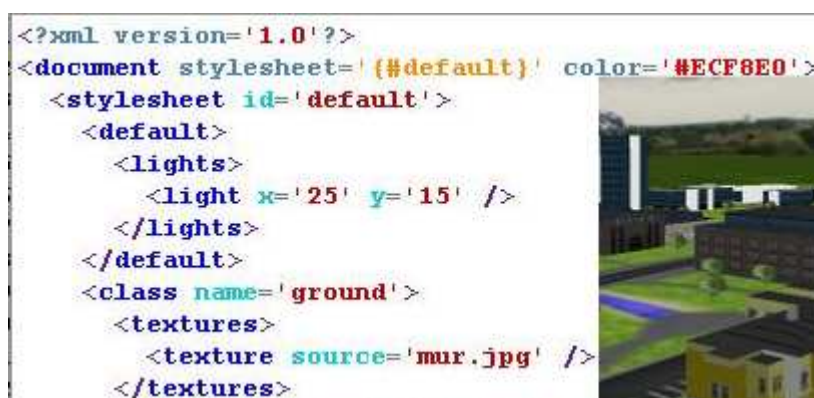


Figura 18 – 3DMLW

Fonte: http://www.freewarefiles.com/3DMLW_program_45880.html

3DMLW (3DMLW, 2008) é uma tecnologia *open source* para criação e exibição de conteúdo 3D e 2D na web através de *browsers*. A construção do conteúdo é feita através de uma linguagem, que leva o nome da própria tecnologia, baseada em XML (Extensible Markup Language), sendo, por isso, de fácil entendimento e programação. É possível também criar scripts baseado na linguagem *LUA*, para criação de conteúdo dinâmico e interativo. A exibição é feita através de um player próprio para a linguagem, que está disponível para a maioria dos *browsers* existentes nos ambientes Windows e Linux.

Nativamente ele trabalha com os tipos de arquivos 3D mais comuns. Isso facilita um pouco o uso da linguagem, já que possibilita construir objetos 3D em alguns dos mais utilizados programas de modelagem.

Internamente ele possui uma *engine* de física, através da implementação da *Newton Game Physics*. Apesar de não ser um requisito procurado pelo projeto, é um ponto interessante dentro da tecnologia.

No entanto, 3DMLW Figura 18, é uma tecnologia muito recente que, em menos tempo ainda, se tornou *open source*. A última notícia de desenvolvimento era uma nova versão da *engine* completa, demonstrando que não se apresenta como um projeto sólido e, por isso, foi descartada como ferramenta de construção do projeto.

4.4 UNITY

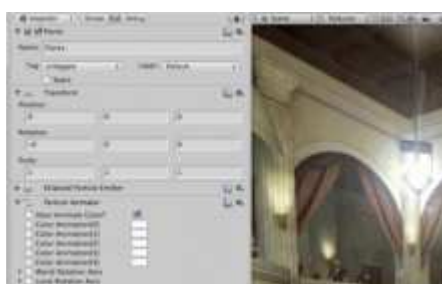


Figura 19 – Unity

Fonte: <http://www.universegames.com.br/blog/unity-3d/>

O Unity (UNITY, 2009) (Figura 19) é um ambiente de desenvolvimento voltado principalmente para jogos. Possui uma *engine* gráfica de alta qualidade e trabalha independente de qualquer outra ferramenta, com a possibilidade de ser executado em sistemas usando Linux e até mesmo em equipamentos como *IPhone*. A portabilidade foi um dos pontos considerados para que fosse utilizado no projeto.

Porém, não é uma ferramenta gratuita e seu produto final não utiliza nenhum padrão. Por estas razões, esta tecnologia foi descartada.

4.5 CROQUET

Croquet (The Croquet Consortium, 2009) (Figura 20) é um kit de desenvolvimento *open source* para construção de ambientes colaborativos. Utiliza redes baseadas em usuários, ao

contrário do modelo cliente-servidor, fazendo com que não seja necessária a existência de um servidor para que a aplicação seja executada.

É um software que enfatiza a parte de sincronização entre usuários. Para fazer com que o programa se adequasse ao projeto, muito tempo seria alocado, além da necessidade de se estudar a linguagem *Squeak*, utilizada na programação do software. Por estas razões, esta ferramenta foi descartada.



Figura 20 – Croquet

Fonte: <http://nsuslovi.blogspot.com/2009/07/live-and-collaborative-coding-in.html>

4.6 O3D

O3D (GOOGLE, 2009b) (figura 21) é uma API open-source *JavaScript* criada pelo Google para criar aplicações interativas em 3D a serem executadas nos navegadores. Isto é possível instalando um plug-in no *browser* para que ele seja capaz de executar as funções *javascript* da API.

O3D encontra-se num estágio inicial de desenvolvimento e propagação. No entanto, existem discussões de que a Google deveria apoiar o uso do *X3D* (Web3D Consortium, 2008a), que é um padrão já estabelecido e que pode ser usado para os mesmos aplicativos que o *O3D* se propõe a construir. Pode ser que isto aconteça no futuro e que todo o investimento feito para aprender *O3D* seja descartado.

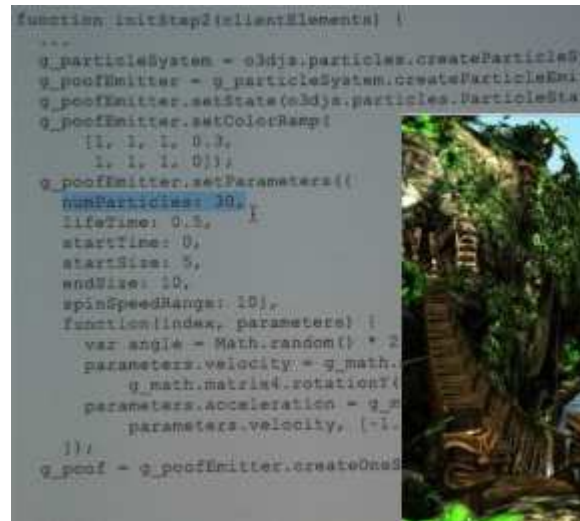


Figura 21 – O3D

Fonte: <http://www.activeclick.com.br/blog/o3d-google-da-um-passo-para-a-web-3d/>

4.7 X3D

```

<Scene>
  <Group DEF="mundo">
    <Background skyColor='1 1 1' />
    <Group DEF='BASE'>
      <Transform DEF='GROUND' translation='0 0 -75'>
        <Shape>
          <Appearance>
            <ImageTexture repeatS='false'
              repeatT='false' url='./floor.jpg' />
            <TextureTransform scale='1 1' />
          </Appearance>
          <Box size='200 0.5 200' />
        </Shape>
      </Transform>
    </Group>
    <Transform DEF='HOUSE' translation='-15 10 -85'>
      <Group>

```



Figura 22 – X3D

X3D (figura 22) é uma tecnologia para distribuição de conteúdo interativo em 3D, que evoluiu da última versão do VRML (CAREY; BELL; MARVIN, 2009). Ela se estabeleceu como um padrão ISO que provê a um sistema, maneiras de armazenar, resgatar e reproduzir gráficos em tempo real contidos em aplicações. Essa linguagem foi produzida com o objetivo de ser usada em uma gama de dispositivos e aplicativos, de áreas como a engenharia, para visualizações

científicas, apresentações em multimídia, e a própria internet. O motivo da *X3D* ser tão abstrata é que possibilita o desenvolvimento em vários periféricos. Pelo fato de ter integração com a linguagem de marcação *XML*, o *X3D* está apto a sobreviver a uma possível mudança do formato de transmissão de dados pela Internet. Outra grande mudança foi o uso do conceito de componentes e extensibilidade. Tendo apenas um núcleo com as funcionalidades antes encontradas no *VRML97*, o *X3D* pode ganhar novas funcionalidades com a expansão deste núcleo adicionando componentes. Graças a essa arquitetura modular, isso o torna altamente flexível.

Através de um protocolo de manipulação para o grafo de cena, denominado *Scene Access Interface (SAI)* (Web3D Consortium, 2008b), o usuário poderá interagir com o ambiente através de códigos via script ou de aplicativos externos. Isso permite que o *X3D* possa ser integrado com outros aplicativos que sigam as especificações dessa interface. Devido a todas essas características, esta foi a ferramenta escolhida para a construção do sistema.

E, recentemente, com a concretização do padrão *HTML 5*, a *World Wide Web Consortium (W3C)*, entidade que desenvolve padrões para conteúdo da internet, está trabalhando para adicionar suporte 3D ao mesmo. Sabendo disso, a *Web3D Consortium* está apresentando idéias em como integrar *X3D* com o *HTML* (X3D 2009). Para maiores detalhes acesse: http://www.3d-test.com/interviews/x3d_2.htm.

4.8 OUTRAS FERRAMENTAS

Assim como foi feito no capítulo anterior, é importante destacar que outras ferramentas foram surgindo durante a construção do projeto. Mais notavelmente, *JMonkeyEngine* (jMonkeyEngine, 2009) e *UDK* (*UDK - Unreal Development Kit* 2009). Ambas são *engines* focadas na construção de jogos, todavia, podem ser adequadamente usadas para outros propósitos.

5 O SISTEMA

Depois de realizada a pesquisa, o sistema ficou definido da seguinte forma: para a interface visual, a utilização de *X3D*; a programação ficaria a cargo da linguagem Java, sendo esta multi-plataforma e disponibilizando um visualizador *X3D*, através da API *Xj3D* (Web3D Consortium, 2008c), que também integra o *SAI* com a linguagem de programação.

O *SAI* é uma API que possibilita desenvolver aplicativos *X3D* dinamicamente. Este protocolo, que manipula a cena *X3D*, pode tanto manipular o navegador quanto o grafo de cena de duas filosofias diferentes: a interna, através dos script nodes, ou a externa, por meio de uma aplicação. Ele é o método usado para programar, através da linguagem Java, eventos em tempo de execução. O *X3D* ficará incumbido pela saída de dados, no caso, a exibição do ambiente. O Java é o componente de programação, e o *SAI*, a ponte que faz a comunicação entre esses dois elementos.

Para a intercomunicação entre a programação e parte visual, usamos o *Xj3D*, um *browser open source*, que contém bibliotecas de desenvolvimento e faz uso da tecnologia OpenGL. Por ser uma aplicação escrita em Java, o *Xj3D* pode ser integrado em outras aplicações Java e é capaz de aceitar a criação de aplicativos através do “Java Web Start”, para transformá-lo em *applet* e conseguir executar em qualquer máquina que acesse o site, fazendo o *download* automático de todos os recursos necessários.

O livro *X3D: Extensible 3D Graphics for Web Authors*, escrito por Don Brutzman (BRUTZMAN; DALY, 2007) foi usado como guia de programação durante o desenvolvimento do software. Além dele, muitos tutoriais na internet, contendo exemplos de códigos, principalmente da API *Xj3D*, foram consultados.

Neste momento, estávamos aptos a decidir como seria a sintaxe a ser utilizada como entrada de dados para o sistema, como fazer o avatar falar e se era possível utilizar algum padrão que facilitasse a construção do avatar. Estes pontos serão vistos a seguir.

5.1 ARQUIVO DE ENTRADA DE DADOS

O sistema precisa receber um arquivo com o conteúdo da aula para que ele fosse capaz de gerar o ambiente. Isto tornaria possível a customização do ambiente sem precisar programar o código interno do sistema. Sendo assim, decidimos criar uma linguagem, chamada de *Syntax*, que seria utilizada para montar a aula virtual. *Syntax* deriva da linguagem *XML* e necessita ser traduzida para a linguagem da interface gráfica. Para esta tarefa existiam duas APIs Java: a *DOM* (Document Object Model) (W3C, 2008); e a *SAX* (Simple API for XML) (Megginson Technologies Ltd., 2008). Para realizar esta decisão, levamos em conta o benefício que cada API traria, assim como a sua utilização de memória, acarretando um melhor desempenho.

A API *DOM* fornece como resultado de seu processamento uma estrutura em forma de árvore de todo o arquivo processado e o armazena na memória. É interessante quando se quer acessar elementos mais de uma vez ou descobrir relações.

A API *SAX* funciona de maneira diferente. Seu processamento é por demanda. A cada elemento encontrado da sintaxe, a API dispara uma chamada a um método e, a partir dele, o elemento pode ser tratado. Depois dessa chamada, esse elemento é descartado e não mais poderá ser acessado, salvo um recomeço da tradução.

Pela maneira como o sistema foi planejado, a API *SAX* foi escolhida, pois o ambiente 3D é montado durante uma fase inicial, antes da interação com o usuário. Nesta fase, obtemos todas as referências necessárias e armazenamos esses dados de uma forma mais conveniente para o sistema.

5.2 MÉTODO DE FALA DO AVATAR

O conceito de voz está presente no sistema. Para que o avatar seja capaz de falar, o sistema utiliza uma técnica de tradução de texto para voz, também conhecida como *text-to-speech* (TTS). Para tal, o sistema usa o *FreeTTS* (FREETS, 2008), que trabalha através da implementação da *Java Speech API* (JSAPI) (SPEECH API, 2008). A *JSAPI* é uma API que

tem a função, dentre outras, de especificar uma interface para sintetizadores de voz. Já o *FreeTTS* é um sintetizador de voz escrito na linguagem Java, *open source*, que se acoplou ao sistema. Apesar de, internamente, apenas possuir um analisador para a linguagem inglesa, o mesmo pode ser estendido para qualquer outra linguagem, desde que as classes necessárias sejam desenvolvidas. As vozes também podem ser criadas ou utilizadas de projetos como o *MBROLA* (Mbrola Team, 2008). Tudo isto tornaria possível o uso do sistema com o vocabulário nacional, que, no momento, ainda usa o analisador interno do *FreeTTS*.

5.3 H-ANIM

O *X3D* possui um padrão para criação de objetos humanóides. O formato *H-ANIM*, desenvolvido pelo *Humanoid Animation Working Group* (Humanoid Animation Working Group, 2008a), é um padrão internacional que representa uma abstração para modelar humanóides em três dimensões (Humanoid Animation Working Group, 2008b).

A importância da utilização deste padrão é a compatibilidade entre o uso de qualquer humanóide, construído utilizando este mesmo padrão, no nosso programa. Os movimentos criados para um desses humanóides serão interoperáveis com qualquer outro. Isto trouxe uma facilidade no decorrer da construção do software, pois existem movimentos e os próprios humanóides disponíveis para uso, tirando a obrigatoriedade de construir um modelo e implementar todos os movimentos do mesmo.

O formato *H-ANIM* é dividido em quatro tipos, segundo a especificação *H-ANIM* 1.1. Esta classificação, chamada de *Levels of Articulation* (LoA), é que determina a quantidade de *joints* e *segments*, unidades básicas do formato, que o humanóide possui. Quanto maior o nível do *LoA*, mais *joints* e *segments* o humanóide terá, e, conseqüentemente, mais detalhado será. No entanto, quanto maior o detalhe, mais tempo será consumido para animar o humanóide corretamente. Na figura 23 temos a estrutura de um humanóide usando *LoA1*, sendo que os diamantes (texto claro) são os *joints* e os segmentos entre os diamantes (texto escuro) são os *segments*. Tomamos a decisão de utilizar o *LoA1*, pois possui detalhes suficientes para as animações do avatar necessárias no momento e pelo seu baixo custo

computacional, além da existência de modelos humanóides (Ballreich, 2008) prontos para serem usados, sem a necessidade de ter de modelar um.

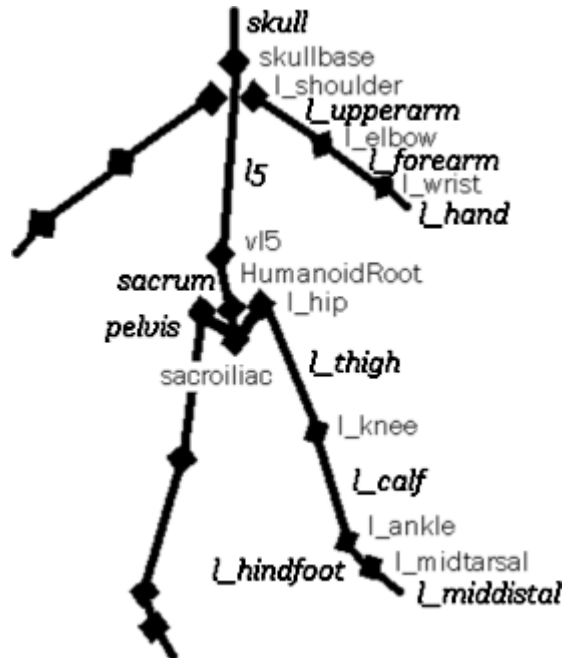


Figura 23 - Estrutura do *LoA1*
 Fonte: Ma e McKevitt (2003)

5.4 LINGUAGEM SYNTAX

Como revelado anteriormente, o sistema precisa fazer leitura de um arquivo que terá a descrição da aula. Para fazer a leitura corretamente, o arquivo deve seguir o seguinte padrão: ele deve iniciar com a *tag* `<syntax>`, indicando o início do arquivo. Da mesma forma, deve terminar com a *tag* `</syntax>`.

Logo após, deverá conter a *tag* `<head>`. Dentro desta *tag* teremos as definições importantes para a criação do ambiente. A *tag* `<env>` é utilizada para estabelecer as configurações do ambiente. Através do parâmetro *src*, podemos definir o arquivo *X3D* com o ambiente construído. E com o parâmetro *imgdef* definimos qual objeto dentro do arquivo *X3D* será o quadro. O quadro tem a função de mostrar as figuras determinadas pela *tag* ``, a ser discutida brevemente.

A tag *<prof>* contém as configurações do avatar do professor. O parâmetro *src* serve para definir o arquivo *X3D* que terá o avatar dentro das especificações *H-ANIM LoAI*. Com o parâmetro *def*, indicamos o objeto do arquivo especificado em *src* para que o sistema consiga controlar o avatar corretamente. E utilizando os parâmetros *pos*, *dim* e *dir* estaremos configurando a posição, tamanho e direção inicial do avatar.

Com tag *<x3dobj>* definimos objetos complexos a serem utilizados no ambiente. Além dos mesmos parâmetros utilizados na tag *<prof>*, ele ainda possui o parâmetro *visible*, que indica se o objeto deve aparecer assim que o cenário seja construído.

Por último, temos a tag *<stu>*, que serve para inserir um novo avatar na cena, além do professor. A idéia desta tag é introduzir os alunos na sala de aula. Eles podem ser controlados da mesma forma que o professor será controlado, através de animações e falas.

Depois de estabelecer todas essas variáveis da tag *<head>*, podemos prosseguir com a construção do arquivo utilizando basicamente as seguintes tags. A tag *<talk>* é utilizada para fazer com que o sistema transforme o texto escrito no parâmetro *value* em som. Para diferenciar a fala entre a do professor e a dos alunos, o parâmetro *target* pode ser usada, passando um valor que represente a voz do avatar que estará falando. No caso, passando o valor *student* fará com que o sistema simule a voz de um aluno ao reproduzir a frase.

Podemos usar a tag *<motion>* para definir um gesto que o avatar irá executar. Este gesto é estabelecido pelo parâmetro *anim*. Podemos personalizar este gesto passando informações extras, através do parâmetro *params*. Os parâmetros *pos*, *dim* e *dir* também estão presentes, para especificar a posição, tamanho e direção final do avatar. Indicamos a duração desse gesto com o parâmetro *dur*. Podemos especificar o avatar que irá executar esta animação através do parâmetro *target*, que deve ser compatível com o valor do parâmetro *def* das tags *<prof>* e *<stu>*.

As tags *<talk>* e *<motion>* possuem também os parâmetros *async* e *pause*. *Async* serve para indicar que o sistema não precisa esperar o comando atual terminar sua execução para continuar com a execução do próximo. *Pause* força uma pausa por um determinado tempo antes de executar o próximo comando; com a tag **, podemos colocar figuras no formato

.jpg ou *.png* para aparecer no objeto definido anteriormente por *imgdef*. Opcionalmente, podemos definir um objeto específico para que essa figura apareça através do parâmetro *def*; e, por fim, podemos, com a tag `<x3dobj>`, manipular o objeto anteriormente definido por esta mesma tag dentro da tag `<head>`, utilizando os parâmetros já conhecidos *pos*, *dim* e *dir*. Além deles, podemos tornar este objeto visível ou não através do parâmetro *visible*. Um exemplo de um arquivo *Syntax* pode ser visto na Figura 24.

```

1  <syntax>
2  <head>
3    <env src="aula.x3d" imgdef="itQuadro" />
4    <prof def="ha_humanoidBody" src="avatarf.x3d" pos="7 1 -120" dim="6 6 6" dir="0 0 0" />
5    <x3dobj src="quimica.x3d" def="cholesterol" visible="false" pos="7 10 -110" />
6  </head>
7  <motion async="true" dur="1" anim="StretchArm" params="leftArm:true;angle:5" />
8  <talk async="true" value="today we are going to have some chemistry class" />
9  <talk value="more precisely about organic chemistry, organic compounds." />
10 <talk value="but first, lets meet our protagonist, mister carbon" />
11 
12 <talk value="mister carbon is the chemical element with symbol c, atomic number 6 and
a member of group 14. it is a nonmetallic and tetravalent element" />
13 <motion dur="1" anim="Stand" dir="0 90 0" />
14 <motion dur="2" anim="Walk" pos="30 1 -120" />
15 <motion dur="1" anim="Stand" dir="0 0 0" />
16 <motion async="true" dur="1" anim="StretchArm" params="angle:119" />
17 <talk value="there you are, mister carbon, right above my wrist" />
18 <talk value="now, let me show you an organic compound" />
19 <motion async="true" dur="1" anim="Stand" />
20 <talk value="how about the cholesterol?" />
21 <x3dobj def="cholesterol" visible="true" />
22 <motion dur="1" anim="Stand" dir="0 -45 0" />
23 <talk value="spinning cholesterol, just look at this! so amazing!" pause="3000" />
24 <talk value="see you in the next class! bye bye" />
25 </svntax>

```

Figura 24 – Exemplo de um arquivo *Syntax*

Um melhor detalhamento do arquivo mostrado na Figura 24, encontra-se adiante, na seção 5.6.

5.5 DETALHES DO SISTEMA

A figura 25 revela resumidamente como o projeto funciona atualmente de uma maneira geral. A linguagem de programação *Java* para realizar a conexão entre os componentes, a linguagem *Syntax*, abstrata, que diz como a aula deve ser construída, o padrão *X3D* para exibição do visual, um padrão de modelagem humana para gestos e construção de avatares (*H-Anim*), e uma ferramenta de conversão de texto em voz, para que o avatar possa dialogar no ambiente virtual.

Toda essa infra-estrutura debatida no decorrer do capítulo 5 facilitou o trabalho de implementação do programa. A classe principal do *DC3DR*, *X3DSwing*, contém a parte de inicialização da API *Xj3D*, realizando a criação da janela no desktop ou do *applet* no *browser*. Nesta janela, uma URL do arquivo apontando para a aula no formato *Syntax* é fornecida pelo usuário. A classe *SyntaxInterpreter* então é criada.

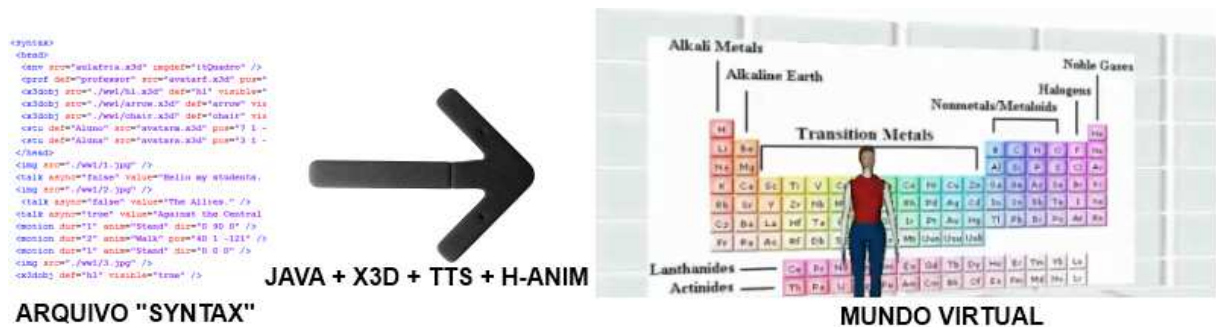


Figura 25 – Fluxo final do projeto

Usando a API *SAX*, *SyntaxInterpreter* avalia e extrai todas as informações necessárias deste arquivo para gerar o ambiente virtual, através do método *parse()*. Depois que toda a informação da aula está na memória, *SyntaxInterpreter* chama a classe *SceneEngine*, que é a classe responsável por gerar os objetos na cena a partir das informações coletadas do arquivo. *SceneEngine* faz uso da janela criada por *X3DSwing* e, através da API *Xj3D*, começa a preparar a cena para ser visualizada, por meio do método *setup()*. Durante esta fase, as classes *MovementEngine* e *SpeechEngine* são instanciadas e ficam aguardando comandos para serem executados. No final do método *setup()*, a cena já está completamente pronta, com todos os avatares já criados visualmente, juntamente com o ambiente. A lista de comandos de voz, de movimentos e do aparecimento dos objetos da cena está armazenada em *SceneEngine* em um único vetor, na mesma ordem na qual estava no arquivo *Syntax* fornecido.

Por fim, a última classe que *SceneEngine* chama é a classe *TouchTimeStarter*. Esta classe ficará escutando um clique em algum objeto da cena, o sensor da cena. No nosso caso, quando *TouchTimeStarter* é declarado, o objeto que ela começa a escutar é o avatar do professor. Assim que o usuário clicar no avatar do professor, *TouchTimeStarter* executa o método *readableFieldChanged()*, que consulta, em ordem, os comandos armazenados por *SceneEngine* e dispara o método interno de execução de cada um deles. Na Figura 26 temos o

diagrama de sequência do sistema, sendo que a criação do sensor, *TouchTimeStarter*, é feita pelo *SceneEngine*, que é o que de fato ocorre no sistema. O *TouchTimeStarter* apenas fica escutando o clique do usuário.

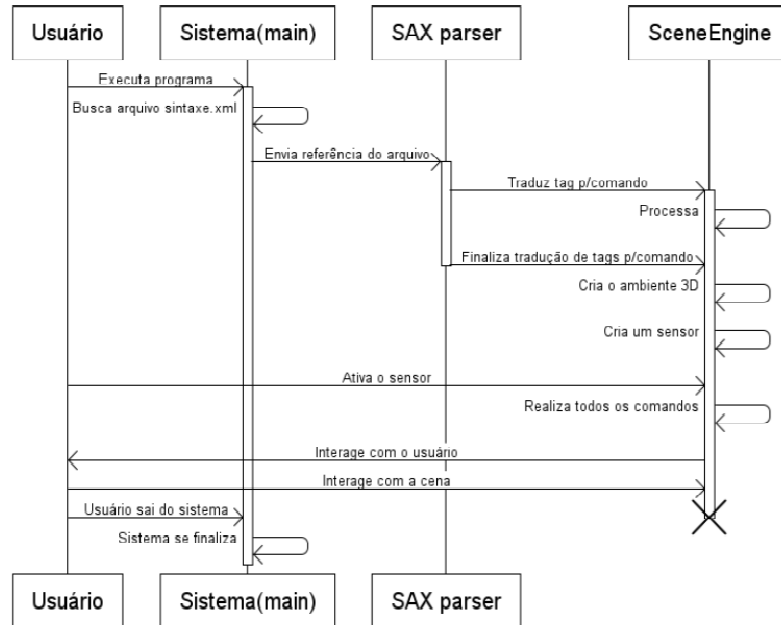


Figura 26 - Diagrama de Sequência do Sistema

Os comandos de voz utilizam a classe *SpeechEngine* para executar. O método *say()* executa o comando de voz de modo síncrono, que é o comportamento natural da API *FreeTTS* e o método *run()* o faz de modo assíncrono, apenas executando o método *say()* em uma *thread*.

Os comandos de movimento fazem uso da classe *MovementEngine*. Internamente, esta classe possui métodos que descrevem o comportamento de cada *joint* e *segment* de acordo com o *H-ANIM LoAI*. O valor passado pelo parâmetro *anim* na verdade é o nome do método dentro desta classe. Quando um comando de movimento é executado, *MovementEngine* entra em ação, verificando o parâmetro *anim* e chamando um método que possua o mesmo valor. Através do parâmetro *params*, ele passa os valores separados para cada um dos parâmetros do próprio método que ele chamou.

Como o comportamento da manipulação da cena é de forma assíncrona, *MovementEngine*, diferente de *SpeechEngine*, não possui um método para executar sincronamente e um segundo para executar assincronamente. Ao invés disso, ele verifica o comportamento do comando de movimento e sinaliza uma flag que *SceneEngine* possui acesso. Quando esta flag está ativada,

SceneEngine só irá executar o próximo comando quando *MovementEngine* avisar que terminou o comando que estava executando.

Os comandos de aparecimento são executados pelo próprio *SceneEngine*. São comandos simples, que apenas manipulam com um parâmetro de visibilidade do próprio X3D.

5.6 EXEMPLO DE USO

O sistema começa requisitando um arquivo seguindo os padrões da linguagem *Syntax* para poder iniciar a construção do ambiente. Depois de realizar a tradução de cada elemento para seus respectivos comandos da interface gráfica, utilizando a API *SAX*, o sistema abre a janela, que é na verdade o *browser X3D* fornecido pela API *Xj3D*, com o cenário e o avatar especificados pelo arquivo *Syntax*.

Para um melhor entendimento da linguagem, vamos exemplificar o seu uso com a seguinte situação: o conteudista deseja disponibilizar uma breve aula de química e mostrar a tabela periódica e uma molécula que se relacione com o assunto da aula. Ele elabora o arquivo *Syntax* mostrado na Figura 24.

As linhas 3 e 4 configuram o ambiente e o avatar do professor respectivamente. Na linha 5 temos a inclusão de um objeto complexo, que neste caso é a estrutura química do colesterol. Já que ele não deve aparecer assim que a aula comece, *visible="false"* foi definido. A partir da linha 7, temos a sequência de comandos que ocorrerá na aula. Nesta mesma linha, podemos ver a *tag <motion>* em ação. O gesto *StretchArm* está definido dentro do sistema e faz com que o avatar levante o braço. É possível indicar qual dos braços e o ângulo para que ele erga o braço através do parâmetro *params*. Neste caso, queremos erguer o braço esquerdo e mover ele apenas cinco graus. O fato de que este comando possua *async="true"* faz com que o comando da linha 8 seja executado imediatamente. No entanto, o mesmo não acontece com o comando da linha 9. Apesar de a *tag <talk>* da linha 8 estar com *async="true"*, a linha 9 possui outra *tag <talk>*. Quando o sistema se depara com uma situação dessas, ele impede que a outra *tag* execute, para não gerar comportamentos inválidos, como ter duas falas ao mesmo tempo, ou dois comandos de deslocamento. Na linha 11 estamos estabelecendo a imagem, vide Figura 27, a ser mostrada na cena.

Alkali Metals										Alkaline Earth										Transition Metals										Nonmetals/Metalloids										Halogens										Noble Gases									
H																				He																																							
Li	Be																	B	C	N	O	F	Ne																																				
Na	Mg																	Al	Si	P	S	Cl	Ar																																				
K	Ca	Sc	Ti	V	Cr	Mn	Fe	Co	Ni	Cu	Zn	Ga	Ge	As	Se	Br	Kr																																										
Rb	Sr	Y	Zr	Nb	Mo	Tc	Ru	Rh	Pd	Ag	Cd	In	Sn	Sb	Te	I	Xe																																										
Cs	Ba	La	Hf	Ta	W	Re	Os	Ir	Pt	Au	Hg	Tl	Pb	Bi	Po	At	Rn																																										
Fr	Ra	Ac	Rf	Db	Sg	Bh	Hs	Mt	Uun	Uuu	Uub							Uuq	Uub																																								
Lanthanides		Ce	Pr	Nd	Pm	Sm	Eu	Gd	Tb	Dy	Ho	Er	Tm	Yb	Lu																																												
Actinides		Th	Pa	U	Np	Pu	Am	Cm	Bk	Cf	Es	Fm	Md	No	Lr																																												

Figura 27 – Arquivo *ptable.jpg*

A aula termina depois de o sistema executar o último comando, na linha 24. O usuário então pode encerrar a aplicação. A demonstração desta aula, figura 28, dura aproximadamente 50 segundos e pode ser visualizada através de: <http://bit.ly/bgDKn5>. Outro exemplo de uma aula de matemática pode ser visto através de: <http://bit.ly/bU3iVw>.

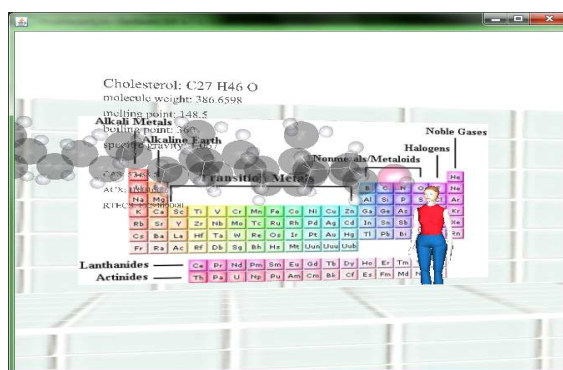


Figura 28 – Sistema em execução

A linguagem *Syntax* é abstrata. Apesar de o exemplo citado neste texto ser uma aula de química, a linguagem independe do conteúdo, seja ele de matemática, português, história ou qualquer outro. A linguagem foi construída com o objetivo de permitir que componentes, como o avatar do professor, a sala de aula e os requisitos necessários para exibir o conteúdo da aula, entre outros, fossem personalizáveis e se adequassem ao conteúdo a ser transformado. Com a troca de um parâmetro, podemos alterar o ambiente virtual, no caso mudando o parâmetro *src* da tag `<env>`, ou mudar o professor, alterando o parâmetro *src* da tag `<prof>`.

Dado o caráter lúdico da ferramenta, avalia-se que este software é voltado mais para alunos de 2º grau. Mesmo por que, apesar de todas as vantagens que a RV oferece e que os educadores já avaliaram, no final, o que importa é a relação do aluno com o assunto. Existem ocasiões que podem ser melhores absorvidas apenas fazendo uma simples leitura do conteúdo

produzido pelo professor. Mas, no geral, a ferramenta possui grande utilidade, por criar uma opção para os professores transmitirem melhor o conteúdo para seus alunos.

6 CONCLUSÃO

Este trabalho conseguiu demonstrar o desenvolvimento de uma ferramenta que tem a proposta de criar um ambiente virtual para representar um local de aprendizado, criar um avatar, representando um professor e fazê-lo falar, tudo isso em um ambiente 3D, para que os usuários desse aplicativo tenham mais uma forma de consultar e aprender o conteúdo disponibilizado por professores. A ferramenta utiliza padrões estabelecidos e fornece maneiras de introduzir o conteúdo do professor, bem como configurar o ambiente virtual. O software foi programado de tal forma que, caso haja necessidade, poderá ter novas funcionalidades incluídas a qualquer momento, como a inclusão de novas *tags*, mudanças nas vozes e nas animações do avatar e até a inclusão de legendas ou um avatar que tenha função de um intérprete de Língua Brasileira de Sinais (LIBRAS).

No passado, o conteúdo estava disponível em livros. Hoje, podemos ter esse mesmo conteúdo em meios digitais. E, com o uso deste programa, o usuário poderá visualizá-lo e escutá-lo. O software ainda precisa ser testado na prática com alunos e conteudistas para que ele seja avaliado. Por hora, conseguimos provar conceitualmente que é possível auxiliar professores em novas formas de transmitir conhecimento.

De acordo com Heerema (2009), um estudo sobre fumantes foi conduzido através de um jogo e descobriu-se que os fumantes que jogavam o jogo tinham maiores chances em relação aos que estavam em grupos de controle. É possível concluir que, mesmo sendo virtuais, estas ferramentas estão tendo a capacidade de influenciar o real.

Outro detalhe importante a ser destacado é o crescente uso da educação à distância. Segundo a Associação Brasileira de Educação a Distância (Abed) em 2000, 13 cursos superiores reuniram 1.758 alunos. Já em 2008, eram 1.752 cursos com 786.718 alunos. Os formandos em cursos de educação a distância ainda sofrem preconceitos e dificuldades em conseguir emprego, conforme (MARTINS; MOÇO, 2009). Mas, com o crescente aumento desses cursos, este quadro tende a ser revertido, demonstrando que ferramentas que tenham foco neste tipo de ensino não só estarão contribuindo para esta situação, como também se preparando para as demandas da educação no futuro.

6.1 TRABALHOS FUTUROS

Como sugestões para trabalhos futuros, poderemos acrescentar interatividade no software, implementando no avatar do professor funções para dialogar com o usuário e criar até exercícios com perguntas e respostas dentro do ambiente. Para melhorar a experiência com o usuário, pode-se tornar o sistema web-based, transformando-o num *applet* e integrando-o com sistemas onlines de tutoria. Para tornar-se um sistema colaborativo, pode-se fazer com que cada usuário que acesse o sistema consiga enxergar os outros alunos que já estão conectados, assistindo suas respectivas aulas.

Fora do domínio da computação, deve-se avaliar esta proposta realizando experimentos didático-pedagógicos, por profissionais da área de educação.

REFERÊNCIAS

3DMLW – Home. Disponível em: <<http://3dmlw.com>>. Acesso em: 11 jul. 2008.

3DXplorer: 3D web accessible to all. Disponível em: <<http://www.3dexplorer.com>>. Acesso em: 10 jul. 2008.

Adobe Systems Incorporated. **Adobe Flash CS4 Professional**. Disponível em: <<http://www.adobe.com/br/products/flash>>. Acesso em: 29 nov. 2009.

ALICE.org. Disponível em: <<http://www.alice.org/>>. Acesso em: 15 set. 2009.

AUKSTAKALNIS, Steve.; BLATNER, David. **Silicon mirage** : the art and science of virtual reality. Berkeley, CA: Peatchpit Press, 1992.

AVATAR. Disponível em: <<http://bit.ly/azO4xZ>>. Acesso em: 02 jan. 2010.

BALLREICH, Cindy. **Nancy - 3D Model 3Name3D**. 1997. Disponível em: <http://www.ballreich.net/vrml/h-anim/nancy_h-anim.wrl>. Acesso em: 11 jul. 2008.

BIOCCA, Frank ; LEVY, Mark. **Communication in the age of virtual reality**. [1995]. Disponível em: <<http://portal.acm.org/citation.cfm?id=207922>>. Acesso em: 22 mar. 2010.

BRAGA, Mariluci. Realidade virtual e educação. **Revista de Biologia e Ciências da Terra** v.1, n. 1, 2001. Disponível em: <<http://bit.ly/d6IHDy>>. Acesso em: 22 nov. 2009.

BRUTZMAN, D., DALY, N. **X3D**: Extensible 3D graphics for web authors. São Francisco: Elsevier, 2007.

CAREY, Rikk.; BELL, Gavin.; MARVIN, Chris. **VRML history homepage**. Disponível em: <<http://bit.ly/9MYSRg>>. Acesso em: 3 set. 2009.

CHAVES, Eduardo. **Tecnologia na educação**: conceitos básicos. 1999. Disponível em: <<http://bit.ly/dBqZys>>. Acesso em: 20 nov. 2009.

DUKE UNIVERSITY. **Cobalt**. Disponível em: <<http://www.duke.edu/~julian/Cobalt/Home.html>>. Acesso em: 3 jul. 2009.

EHRlich, Justin; MILLER, James. A virtual environment for teaching social skills: AViSS. **IEEE Computer Graphics and Applications**, v. 29, n. 4, p. 10-16, Jul./Aug., 2009.

EXIT Reality. Disponível em: <<http://www.exitreality.com/>>. Acesso em: 10 jul. 2008.

FREETTS 1.2 - A speech synthesizer written entirely in the Java programming language. Disponível em: <<http://freetts.sourceforge.net/docs/index.php>>. Acesso em: 11 jul. 2008.

FREE PATENTS ONLINE. Disponível em: <<http://bit.ly/cSe4kG>>. Acesso em: 02 jan. 2010.

GOOGLE. **Lively-3D Avatars and Rooms**. Disponível em: <<http://www.lively.com/goodbye.html>>. Acesso em: 20 abr. 2009a.

GOOGLE. **O3D API - Google Code**. Disponível em: <<http://code.google.com/apis/o3d/>>. Acesso em: 8 ago. 2009b.

HALL, Trish. Virtual reality takes its place in the real world. **New York Times**, jul. 1990.

HANCOCK, D. - Viewpoint: virtual reality in search for middle ground. **IEEE Spectrum**, v. 32, n. 1, jan. 1995.

HEEREMA, Matt. **Virtual reality helps some smokers quit**. Disponível em: <<http://bit.ly/1CPAtA>>. Acesso em: 12 nov. 2009.

Humanoid Animation Working Group. **Humanoid Animation Working Group**. Disponível em: <<http://www.hanim.org/>>. Acesso em: 11 jul. 2008a.

Humanoid Animation Working Group. **ISO/IEC FCD 19774:200x – Introduction**. Disponível em: <<http://bit.ly/8Z2mqt>>. Acesso em: 11 jul. 2008b.

JAVA SE Desktop Technology. Disponível em: <<http://java.sun.com/javase/technologies/desktop/java3d/>>. Acesso em: 5 abr. 2009.

JAVA Technology. Disponível em: <<http://www.sun.com/java/>>. Acesso em: 17 ago. 2008.

jMonkeyEngine.com. Disponível em: <<http://www.jmonkeyengine.com/index.php>>. Acesso em: 29 nov. 2009.

LEVY, Pierre. **O que é o virtual?**. São Paulo: Editora 34, 1996.

LIMA, Claudia Regina Uchôa. O estado da arte da realidade virtual. **Informática na Educação: teórica e prática**. v. 1, n. 2, 1998. Disponível em: <<http://bit.ly/aLfJFX>>. Acesso em: 02 jan. 2010.

MA, Minhua; McKEVITT, Paul. **Building character animation for intelligent storytelling with the H-anim standard**. In: EUROGRAPHICS IRELAND CHAPTER WORKSHOP 2003, M. McNeill (Ed.), Coleraine, Northern Ireland, April. 2003, p. 9-15.

MARANGONI, Barone; PRECIPITO, Barilli. **Reconhecimento e sintetização de voz usando Java speech**. [2006]. Disponível em: <<http://bit.ly/ccDFTS>>. Acesso em: 22 mar. 2010.

MARTINS, Ana Rita; MOÇO, Anderson. Educação a distância: vale a pena entrar nessa? V. **Revista Nova Escola**. São Paulo, v. 24, n. 227, p. 52- 59, nov. 2009.

MCLELLA, Hillary. **Virtual Realities**. Disponível em: <<http://bit.ly/d1cf1t>>. Acesso em: 22 mar. 2010.

Meggison Technologies Ltd. **Simple API for XML**. Disponível em: <<http://www.meggison.com/downloads/SAX/>>. Acesso em: 11 jul. 2008.

MERCADO, Luís Paulo Leopoldo; LIRA, Mayara Teles Viveiros; LIRA, Cíntia Silver. **Educação a distância nas teses e dissertações dos cursos de pós-graduação em educação brasileiros no período 1998-2007**. [2008]. Disponível em: <<http://bit.ly/d8idYD>> . Acesso em: 05 dez. 2009.

MOORE, Michel ., KEARSLEY, Greg. **Distance education: a systems view**. Belmont (USA): Wadsworth Publishing Company, 1996. 290 p.

MORAN, José. **Novas tecnologias e o re-encantamento do mundo**. [1995]. Disponível em: <<http://bit.ly/b9ojSc>>. Acesso em: 22 mar 2010.

MORIE, Jacquelyn Ford. Inspiring the future: merging mass communication, art, entertainment and virtual environments. **ACM SIGGRAPH Computer Graphics**. v. 28, n.2, p. 135-138, May, 1994.

ROSA, Paulo Ricardo da Silva, **O que é ser professor?: premissas para a definição de um domínio da matéria na área do ensino de ciências**. Disponível em: <<http://bit.ly/aqpXua>>. Acesso em: 22 mar. 2010.

SAHIB, Shahrin. **Multimedia in the perspective of Mathematical Modelling: Past, Present and Future**. Disponível em: <<http://bit.ly/cdJHwz>>. Acesso em: 22 mar. 2010.

SCAVETTA, Domencio; LAUFFER, Roger. **Texte, hipertexte, hipermedia**. Paris: Intro, 1997.

SECOND Life. **Virtual worlds, avatars, 3D chat, online meetings - Second Life Official Site**. Disponível em: <<http://secondlife.com/>>. Acesso em: 10 jul. 2008.

SOUSA, Valdivino Alves de. **História da educação**. Disponível em: <<http://bit.ly/bZYkgt>>. Acesso em: 22 mar. 2010.

SOUZA, Luiz Augusto Teles. **Ensino a distância: mudando conceitos**. Campinas, 2005. Disponível em: <<http://bit.ly/csUmQ0>>. Acesso em: 20 nov. 2009.

SPEECH API. Disponível em: <<http://java.sun.com/products/java-media/speech/>>. Acesso em: 11 jul. 2008.

SUTHERLAND, Ivan. **A head-mounted three dimensional display**. Disponível em: <<http://portal.acm.org/citation.cfm?id=1476589.1476686>>. Acesso em: 22 mar. 2010.

The Croquet Consortium. **Main Page - Croquet Consortium**. Disponível em: <http://www.opencroquet.org/index.php/Main_Page>. Acesso em: 5 abr. 2009.

THE MBROLA Project Homepage. Disponível em: <<http://tcts.fpms.ac.be/synthesis/mbrola.html>>. Acesso em: 11 jul. 2008.

THERE - The online virtual world that is your everyday hangout. Disponível em: <<http://www.there.com/>>. Acesso em: 10 jul. 2008.

TRINDADE, Carlos; FIOLEAIS, Carlos. **Concepção de ambientes virtuais**: descrição e apresentação de um caso prático. Disponível em: <<http://bit.ly/9tO29p>>. Acesso em: 02 jan. 2010.

UDK - Unreal Development Kit - Epic Games. Disponível em: <<http://www.udk.com/>>. Acesso em: 29 nov. 2009.

UNITY: Game Development Tool. Disponível em: <<http://unity3d.com>>. Acesso em: 29 nov. 2009.

VIRTUAL REALITY. **Virtual reality, tele-perception and tele-existence**. Disponível em: <<http://shifz.org/tbs/doc/tbsgen.htm>>. Acesso em: 22 mar. 2010.

W3C. **Document object model (DOM)**. Disponível em: <<http://www.w3.org/DOM/>>. Acesso em: 11 jul. 2008.

WEB.alive. Disponível em: <<http://estore.projectchainsaw.com/WaStore/>>. Acesso em: 10 jul. 2008.

Web3D Consortium. **X3D for Developers**. Disponível em: <<http://www.web3d.org/x3d/>>. Acesso em: 11 jul. 2008a.

Web3D Consortium. **Extensible 3D (X3D), ISO/IEC FCD 19775-2.2:200x, 4 Concepts**. Disponível em: <<http://bit.ly/aweDN0>>. Acesso em: 11 jul. 2008b.

Web3D Consortium. **The Xj3D project**. Disponível em: <<http://www.xj3d.org/>>. Acesso em: 11 jul. 2008c.

X3D a write once, render anywhere, anytime 3D graphics format. Disponível em: <<http://bit.ly/6goLXj>>. Acesso em: 28 dez. 2009.

YOUNGBLUT, Christine. **Educational uses of virtual reality technology**. [1998]. Disponível em: <<http://en.scientificcommons.org/17124247>>. Acesso em: 22 mar. 2010.