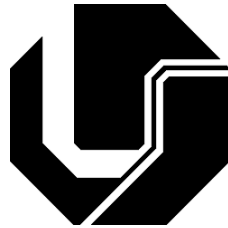


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**O Uso de Técnicas de Realidade Virtual Distribuída
Para Suporte a Jogos Educacionais Multi-Jogadores
na Internet**

Orientador: Edgard Lamounier Junior, PhD

Co-Orientador: Alexandre Cardoso, Dr.

Orientando: Everton Silva de Souza

SETEMBRO

2008

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

EVERTON SILVA DE SOUZA

O Uso de Técnicas de Realidade Virtual Distribuída Para Suporte a Jogos Educacionais Multi-Jogadores na Internet

Dissertação apresentada ao Curso de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como exigência parcial para obtenção do título de Mestre em Ciências.

Área de Concentração: Computação Gráfica /
Realidade Virtual

Edgard Lamounier Júnior, PhD (UFU) - Orientador

Alexandre Cardoso, Dr. (UFU) - Co-Orientador

Marcos Wagner de Souza Ribeiro, Dr (ULBRA-GO)

Keiji Yamanaka, PhD (UFU)

UBERLÂNDIA

2008

O Uso de Técnicas de Realidade Virtual Distribuída Para Suporte a Jogos Educacionais Multi-Jogadores na Internet

Everton Silva de Souza

Dissertação apresentada à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciências.

Prof^o Edgard Lamounier Júnior
Orientador

Prof^o Darizon Alves de Andrade
Coordenador do Curso

Dados Internacionais de Catalogação na Publicação (CIP)

S729u Souza, Everton Silva de, 1981-
O uso de técnicas de realidade virtual distribuída para suporte a
jogos educacionais multi-jogadores na internet / Everton Silva de
Souza. - 008.
132 f. : il.

Orientador: Edgard Lamounier Junior.
Dissertação (mestrado) – Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Realidade virtual - Teses. I. Lamounier Júnior, Edgard, 1964-
II. Universidade Federal de Uberlândia. Programa de Pós-Graduação
em Engenharia Elétrica. III. Título.

CDU: 681.3:007.52

Aos meus pais, Gerson e Creusa Miranda.

Aos meus irmãos Eduardo e Jefferson.

À minha noiva, Cristina Diaz.

Agradecimentos

Agradeço primeiramente a Deus que me permitiu finalizar uma etapa em minha vida.

Ao meu orientador e amigo, Edgard Lamounier Jr., pelo empenho, pela confiança depositada em mim e por mostrar-me sempre o melhor caminho a seguir.

Ao meu co-orientador, Alexandre Cardoso pelo profissionalismo e grande colaboração em todos os momentos.

Aos professores Marcos Wagner, Keiji Yamanaka, pela cordial atenção e pelas contribuições ao projeto.

À Cristina Diaz, que demonstrou uma paciência única ao meu lado, apoiando-me sempre com atenção e carinho.

Ao excelente programa de Pós-graduação em Engenharia Elétrica. À Marli, pelo apoio, presteza e gentil ajuda.

Às crianças e professoras e, em especial a professora Ana Bacha da escola FORTEC, pela colaboração.

Aos colegas Ezequiel, Luciano Ferreira e, em especial, a Marlene Freitas, pelo apoio técnico, troca de conhecimento, amizade e paciência.

Aos demais colegas da Pós-graduação pelo companheirismo.

Aos queridos irmãos Eduardo e Jefferson, pelo apoio.

E, finalmente, aos meus adorados pais, Gerson Gomes e Creusa Miranda, pela compreensão frente ao tempo subtraído ao nosso convívio.

Resumo

Souza, Everton S. *O Uso de Técnicas de Realidade Virtual Distribuída Para Suporte a Jogos Educacionais Multi-Jogadores na Internet*. Faculdade de Engenharia Elétrica – UFU, Uberlândia, 2008, 132p.

Esta dissertação apresenta o desenvolvimento de ambientes virtuais distribuídos por meio de técnicas computacionais que associadas à Realidade Virtual como ferramenta de apoio a projetos de ensino por meio da Internet. Considerando os trabalhos relacionados com o referido tópico, percebe-se que os atuais ambientes virtuais distribuídos não exploram técnicas para melhorar o uso de largura de banda e latência. Diante disto, tem-se como objetivo para esta dissertação, a criação de um jogo 3D usando técnicas da Realidade Virtual multi-jogador para Internet. A principal idéia é demonstrar, por meio de técnicas de RV e sistemas distribuídos, eficiência em interação entre usuários. Por essa razão, foi proposto o estudo de caso LudosTop que é o jogo Quarto, educacional, de apoio ao ensino. O projeto do jogo baseia-se na imagem de um parque. Nesse parque, o usuário tem a possibilidade de jogar uma partida do jogo (o Quarto) contra outro jogador pela Internet. O objetivo do jogo é alinhar sólidos geométricos que possuam atributos comuns, possibilitando adicionar, em determinado momento do jogo, uma terceira ou quarta pessoa que deve observar a partida. O processo de desenvolvimento incluiu análises de jogos para referencial teórico, metodológico e elaboração do software proposto. Para tanto, foi avaliada uma potencial tecnologia para criação e distribuição de ambientes virtuais, que se pode aplicar as técnicas de Realidade Virtual Distribuída é apresentar as vantagens de melhorias na desempenho da comunicação em rede. O trabalho inclui a descrição e implementação das técnicas de Realidade Virtual com métodos da tecnologia Torque Game Engine por meio de bibliotecas C++, OpenGL e Torque Scripting APIs, além de apresentar detalhes de implementação do sistema. O jogo 3D multi-jogadores foi avaliado por crianças do Ensino Fundamental, da 5º série, gerando, por fim, um relatório que indicou contribuição da experiência.

Palavras-chave:

Ambientes Virtuais Distribuídos, Realidade Virtual, Jogos Multi-Jogadores, Jogo Quarto.

Abstract

Souza, Everton. *The Use of Virtual Reality's Techniques to Support of Games Multiplayer Online*. Faculty of Electrical Engineering - UFU, Uberlândia, 2008, 132p.

This thesis presents the development of a Distributed Virtual Environment with the use of computational techniques that, associated to Virtual Reality, as a tool to support project in education to Internet. Considering the related work, you can see that the currents Distributed Virtual Environment did not explore techniques to improve the bandwidth and latency. In face of this, the objective for this dissertation, the development of an online multiplayer 3D game with Virtual Reality techniques. The principal idea is that through Distributed System and VR techniques, can demonstrate efficiency on interaction between players. For this reason, it was proposed the case study LudosTop, it is game Quarto and can aid the educator's activity. Such learning atmosphere returns to mathematical contents and it has, as public, students of the Fundamental Teaching. The design of the software bases on the image of a park. From within of the park, the user has the possibility to play ("Quarto") a match against other player over Internet. The objective of the game is to align geometric solids with common attributes, giving possibility of add, at a certain moment of match; a third or fourth person as observer. The work included the analysis of some educational games, such educational systems served as theoretical and methodological referential for the elaboration of the proposed software. Therefore, a potential technology were evaluated to creation and distribution of Virtual Environment, where it can apply distributed virtual environment (DVE) techniques and make seen advantages in application to improve performance network communication. The work includes description and implementation of virtual reality's techniques with technology Torque Game Engine though C++ Programming Language, OpenGL Library and Toque Scripting API. The 3D Multiplayer Game was tested with children of the fundamental teaching, generating, finally, a report that indicated absolute contribution this experience.

Keywords:

Distributed Virtual Environment, Virtual Reality, Games Multiplayer, and Game Quarto.

Lista de Publicações

A seguir são apresentadas as publicações originadas deste trabalho:

SOUZA, Everton Silva de; SILVA, Luciano; FREITAS, Marlene Roque; LAMOUNIER, Edgard Jr.; CARDOSO, Alexandre. Uso do Middleware CORBA na Distribuição de Ambientes Virtuais para Suportar Jogos Multi-Jogadores. 25th SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, BELÉM-PA, JULHO/2007.

SOUZA, Everton Silva de; SILVA, Luciano; FREITAS, Marlene Roque; LAMOUNIER, Edgard Jr.; CARDOSO, Alexandre. Developing a Multiplayer Online Learning Environment to Web Support with Ajax3d. WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA, ITUMBIARA-GO, NOVEMBRO/2007.

SOUZA, Everton Silva de; SILVA, Luciano; FREITAS, Marlene Roque; LAMOUNIER, Edgard Jr.; CARDOSO, Alexandre. Ludos Top: A Web-based 3D Educational Game Multi-user Online. 6th INTERNATIONAL INFORMATION AND TELECOMMUNICATION TECHNOLOGIES SYMPOSIUM – COMPUTER NETWORKS / REAL TIME / DISTRIBUTED SYSTEMS, BRASÍLIA-DF, DEZEMBRO/2007.

SOUZA, Everton Silva de; CARDOSO, Alexandre; LAMOUNIER, Edgard Jr. Architecture to Support Multiplayer Games through Virtual Environment Distribution. 10th SYMPOSIUM ON VIRTUAL AND AUGMENTED REALITY, JOÃO PESSOA-PB, MAIO/2008.

SOUZA, Everton Silva de; CARDOSO, Alexandre; LAMOUNIER, Edgard Jr. Ludos Top: A Web-Based 3D Educational Game Multi-User to Internet. XIII BRAZILIAN SYMPOSIUM ON MULTIMEDIA AND THE WEB / WORKSHOP ON TOOLS AND APPLICATIONS), GRAMADO-RS, OUTUBRO/2007.

Sumário

1. INTRODUÇÃO	10
1.1. Motivação	10
1.2. Objetivos e Metas.....	12
1.3. Estrutura da Dissertação.....	12
2. REALIDADE VIRTUAL COM SISTEMAS DISTRIBUÍDOS	14
2.1. Introdução	14
2.2. Realidade Virtual.....	14
2.2.1 Terminologia.....	15
2.2.2 Definições	15
2.2.3 Histórico.....	17
2.2.4 Os diferentes níveis de experimentação	20
2.2.5 Imersão, interação e envolvimento.....	22
2.2.6 RV passiva, exploratória ou interativa	23
2.2.7 Aplicações.....	25
2.3. Realidade Virtual Distribuída	26
2.3.1 Conceituação e Caracterização	27
2.3.2 Os componentes da Realidade Virtual Distribuída	30
2.3.3 Comunicação em Rede.....	30
2.3.4 Visões do Usuário de um AVD.....	32
2.3.5 Modelo de Dados	32
2.3.6 Gerenciamento da Computação	34
2.3.7 Comportamento dos Objetos.....	35
2.3.8 Técnicas da Realidade Virtual Distribuída.....	36
2.3.9 O uso de ambiente virtuais distribuídos em EAD	42
2.4. Considerações Finais	43
3. TRABALHOS RELACIONADOS	45
3.1. Introdução	45
3.2. Jogos sem Realidade Virtual sem Distribuição	45

3.2.1	GameKid	45
3.2.2	Dominó Matemágico.....	47
3.3.	Jogos com Realidade Virtual sem Distribuição.....	49
3.3.1	Planeta X.....	49
3.3.2	Supermercado 3D.....	52
3.3.3	Escola TRI-Legal	54
3.3.4	LudosTop	56
3.4.	Jogos com Realidade Virtual com Distribuição	62
3.4.1	EducaTrans	62
3.4.2	Peloton	64
3.4.3	Laboratório Multidisciplinar Virtual em RV.....	66
3.4.4	Laboratório Multidisciplinar Virtual em RA.....	69
3.5.	Tabelas de Comparação dos Jogos Avaliados	71
3.6.	Considerações Finais	72
4.	ARQUITETURA E FUNCIONAMENTO DO SISTEMA	74
4.1.	Introdução	74
4.2.	Tecnologia de Apoio.....	74
4.2.1	Plataforma de Distribuição.....	74
4.2.2	OpenGL API Gráfica	79
4.2.3	<i>Socket</i>	80
4.3.	Arquitetura do Sistema	82
4.3.1	GUI – Interface Gráfica com o Usuário	83
4.3.2	Distribuidor de Eventos para Rede.....	83
4.3.3	Módulo de Jogo Virtual	84
4.3.4	Estudo de Caso e Funcionamento do Sistema.....	86
4.3.5	Hardware requerido para o funcionamento do sistema	94
4.4.	Considerações Finais	94
5.	DETALHES DE IMPLEMENTAÇÃO DO JOGO	95
5.1.	Introdução	95
5.2.	Detalhes de Implementação	95
5.2.1	Objetos Virtuais	96

5.2.2	Algoritmos de Jogadas.....	99
5.2.3	Distribuidor de Eventos.....	104
5.3.	Considerações Finais	106
6.	RESULTADO DA APLICAÇÃO DO SOFTWARE	107
6.1.	Introdução	107
6.2.	Avaliação do Sistema pelos usuários	107
6.3.	Considerações Finais	116
6.4.	Tabelas de Comparação do software proposto (Quarto3D).....	117
7.	CONCLUSÕES E TRABALHOS FUTUROS	119
7.1.	Introdução	119
7.2.	Conclusões	120
7.3.	Trabalhos Futuros	121
7.4.	Considerações Finais	122
	REFERÊNCIAS BIBLIOGRÁFICAS	124

Lista de Figuras

Figura 2-1: Desenho Esquemático de uma CAVE.....	20
Figura 2-2 : Desenho Esquemático de uma ImmersaDesk.....	20
Figura 2-3: Mostra o <i>culling</i> usando o grid (SINGHAL e ZYDA, 1999).....	39
Figura 2-4: Modelo da técnica de particionamento (SINGHAL e ZYDA, 1999).....	40
Figura 3-1: Tela de seleção de personagem no GameKid (BINDA <i>et al</i> , 2005).....	46
Figura 3-2: Interface gráfica do Dominó Matemágico. Fonte (VASCONCELOS <i>et al</i> , 2005)	48
Figura 3-3: Tela do Planeta X (ARAÚJO e CORDENONSI, 2003).....	50
Figura 3-4 : Cenas do Planeta X (ARAÚJO e CORDENONSI, 2003).....	51
Figura 3-5: Tela de Apresentação do Supermercado 3D (POSITIVO, 2008).	53
Figura 3-6: Escola Tri-Legal (VENDRUSCOLO <i>et al</i> , 2005).....	54
Figura 3-7: Ginásio de esporte tridimensional da Escola Tri-Legal (VENDRUSCOLO <i>et al</i> , 2005).....	55
Figura 3-8: Pagina de acesso ao sistema (FREITAS, 2008)	57
Figura 3-9: Botão personagens - linha do tempo (FREITAS, 2008).	58
Figura 3-10: Procedimentos iniciais utilização do sistema (FREITAS, 2008).....	59
Figura 3-11: Opções de apelido e setas para navegar nas opções (FREITAS, 2008).....	60
Figura 3-12: Laptop – ambiente para jogar (FREITAS, 2008).....	61
Figura 3-13: Cenário do EducaTrans (ASSIS <i>et al</i> , 2006).	63
Figura 3-14: Dispositivo de entrada e saída do Peloton (CARRARO <i>et al</i> , 1998).	64
Figura 3-15: Tela do Ambiente virtual de Biologia (RIBEIRO, 2006).	67
Figura 3-16: Tela do Ambiente Virtual de Química (RIBEIRO, 2006).....	67
Figura 3-17: Exemplo de funcionamento do Sistema (SILVA, 2008).	69
Figura 4-1: Configuração das luzes do ambiente virtual.	77
Figura 4-2: Textura do cilindro de seleção.....	78
Figura 4-3: Área do Jogador adversário.	78
Figura 4-4: Versão simplificada do <i>pipeline</i> OpenGL (WRIGHT, 2000).	80
Figura 4-5: Modelo representa a comunicação de sockets e a pilha TCP/IP.	81
Figura 4-6: Arquitetura do software proposto Quarto 3D	82
Figura 4-7: Tela principal de apresentação do jogo.....	83
Figura 4-8: Quarto3D – ambiente para jogar.....	85
Figura 4-9: As 16 peças do jogo, mais 4 x 4 casa (16) no tabuleiro.	87
Figura 4-10: Tabuleiro do jogo Quarto	87
Figura 4-11: LudosTop na versão VRML (FREITAS, 2008).	89
Figura 4-12: Tela de autenticação.....	90
Figura 4-13: Opção de áudio e visualização.....	91
Figura 4-14: Tela ajuda sobre o jogo.....	91
Figura 4-15: Informações dos desenvolvedores.	92

Figura 4-16: Criação de um jogo.....	92
Figura 4-17: Tela do vencedor da partida.....	93
Figura 5-1: Modelagem das peças do jogo Quarto3D.....	95
Figura 5-2: Modelagem do cenário do Jogo.....	96
Figura 5-3: Código de implementação do ambiente virtual do jogo.....	97
Figura 5-4: Posições dos objetos invisíveis.....	98
Figura 5-5: Exemplo de codificação para carregar objetos do 3D Max.....	98
Figura 5-6: Possíveis câmeras do software proposto.....	99
Figura 5-7: Implementação na biblioteca da Torque em C++.....	100
Figura 5-8: Codificação de eventos do mouse em C++.....	100
Figura 5-9: Implementação da função onMouseMove()......	101
Figura 5-10: Exemplificando a execução da função pela Torque.....	102
Figura 5-11: Algoritmo de execução das regras do jogo.....	103
Figura 5-12: Código de verificação do alinhamento do quarto.....	104
Figura 5-13: Codificação de movimentação das peças e câmera.....	105
Figura 5-14: Função de atualização de estados do cliente.....	106
Figura 6-1: A demonstração gráfica sobre o uso do computador.....	108
Figura 6-2: Avaliação geral do software quanto à sua utilidade.....	109
Figura 6-3: Movimentação das peças.....	110
Figura 6-4: Identificação de atributos.....	111
Figura 6-5: Comunicação entre jogadores.....	112
Figura 6-6: Preferência por jogar sozinho ou não.....	113
Figura 6-7: Aprender enquanto joga.....	114
Figura 6-8: Jogabilidade em RV.....	115

Lista das Tabelas

Tabela 3-1: Tabela de comparação entre ambientes virtuais educativos.	71
Tabela 6-1: Dados referentes ao uso do computador pelos alunos pesquisados.	108
Tabela 6-2: Dados referentes à questão: uso do software.	109
Tabela 6-3: Dados referentes à questão: movimentação das peças por meio do mouse.	110
Tabela 6-4: Dados referentes à questão: identificação de atributos.....	111
Tabela 6-5: Dados referentes à questão: comunicação entre jogares via Internet.....	112
Tabela 6-6: Dados referentes à questão: preferência de jogar sozinho ou não.	113
Tabela 6-7: Dados referentes à questão, aprendizado enquanto se joga no computador.....	114
Tabela 6-8: Dados referentes à questão, a jogabilidade em Realidade Virtual.....	115
Tabela 6-9: Tabela de comparativa do software proposto.....	117

Capítulo 1

1. Introdução

1.1. Motivação

Realidade Virtual (RV) é considerada uma tecnologia revolucionária, pois possibilita a simulação de mundos reais e imaginários na tela do computador ou em outros dispositivos, criando no usuário a sensação de presença em um “mundo” virtual (CRUZ-NEIRA, 1992) (CARLSON, 1993).

A Realidade Virtual envolve tecnologias para o desenvolvimento de aplicações, em função de novas possibilidades que oferece para interfaceamento com o uso de dispositivos multi-sensoriais, navegação em espaços tridimensionais, imersão no contexto da aplicação e interação em tempo real (KIRNER, 1995). As pesquisas em novas tecnologias para área de entretenimento e educação vêm crescendo consideravelmente por meio de estudos com a Realidade Virtual. (HEETER, 1989;SCHUMANN, 1999).

Na Realidade Virtual existem várias áreas de aplicações e, ao considerar a crescente popularidade de jogos pela Internet torna-se interessante analisar os principais problemas e soluções relacionadas a sistemas multi-usuários. Na área de Jogos é comum pesquisadores estudarem detalhadamente as relações entre Realidade Virtual ou sistemas distribuídos com a finalidade de demonstrar a complexidade de um ambiente virtual distribuído multi-jogador (CORREIA, 2005). Os sistemas distribuídos em parceria com sistemas de RV possibilitam a criação dos Ambientes Virtuais Distribuídos (AVD) definidos com ambientes virtuais que permitem a participação de diversos usuários ao mesmo tempo (RIBEIRO, 2006, p.1).

A variedade de características que os AVDs devem apresentar, torna esse tipo de sistema cada vez mais complexo, à medida que se aproxima dos ambientes reais. Pois os AVDs possuem uma interação distribuída, permitindo que os usuários compartilhem cada atividade exercida em tempo real entre os usuários, é uma característica essencial dos Ambientes Virtuais Distribuídos.

A pesquisa e projeto de AVDs é uma área bastante fértil, promissora e extremamente estimulante pelos desafios tecnológicos que apresenta e pelo potencial de aplicabilidade. Existem muitas questões a serem melhoradas em temas de aplicações distribuídas interativas, onde necessitam usar a largura de banda e latência disponível sem afetar na interação. A maioria das aplicações 3D distribuídas não possui eficiência na transmissão de dados entre os participantes por meio da internet (MACEDO *et al*, 2005). Além disso, necessitam de conexões rápidas entre os participantes que devem utilizar máquinas com considerável capacidade de processamento. Estas restrições são completamente razoáveis para grupos pequenos dentro de uma única organização, deve ser, mas impróprios para internet.

Desta forma, a Realidade Virtual Distribuída tem potencial para propiciar uma educação como processo de exploração, descoberta, observação e construção de uma nova visão do conhecimento. A utilização de RV com fins educativos tem merecido destaque e tem sido avaliada de forma intensiva nos últimos anos. Conclusivamente, pesquisadores apontam como principais vantagens da utilização de técnicas de RV para fins educacionais (CARDOSO *et al*, 2007).

A educação a distância (EAD) apoiada por Tecnologias de Informação e Comunicação (TIC) permite avanços e possibilidades ainda pouco exploradas. Pesquisadores e educadores estão repensando os modelos pedagógicos para a educação online e novos cenários são decorrentes da mudança de paradigma.

A principal motivação para o desenvolvimento de ambientes virtuais multi-usuários na Web está na possibilidade de transformar a Web de um espaço basicamente de informação para um lugar social, com informação e com uma comunidade de usuários que compartilham essa informação e interagem entre si (KIRNER *et al*, 2000, p.169-179).

Neste sentido o objeto de estudo desta dissertação é a distribuição de ambientes virtuais para Internet, deixando claro que as principais pesquisas concentram-se na melhoria do processo de comunicação entre cliente/servidor, pois apresentam problemas de sincronização devido a diferentes larguras de banda entre usuários da Internet, com empacotamento de entidades, confiabilidade em sistemas distribuídos e falta de controle de autenticação entre cliente – servidor.

1.2. Objetivos e Metas

O objetivo deste trabalho é propor e validar estratégias computacionais associadas à Realidade Virtual e Sistemas Distribuídos para o desenvolvimento de um estudo de caso que seja multi-jogador para Internet e aplicar técnicas de comunicação que promova melhoria no uso de largura de banda e latência, pois possibilitando a criação de um software com interação distribuída com suporte a Internet; e fazer com que tal software possa constituir um auxílio ao educador. Para atingir este objetivo proposto, as específicas metas foram definidas:

- Analisar sistemas com e sem suporte à Realidade Virtual (Estratégia).
- Observar e comentar características das arquiteturas dos softwares analisados.
- Revisar os fundamentos da Realidade Virtual e Sistemas Distribuídos.
- Investigar técnicas de Realidade Virtual, visando criar um software que possibilite a colaboração entre usuários sobre a rede de computadores com suporte à Internet.
- Escolher uma plataforma para o desenvolvimento multi-usuário para desenvolvimento do software proposto.
- Escolher e justificar a escolha do estudo de caso, tomando tal experiência como enriquecimento para a elaboração do sistema proposto.
- Implementar um protótipo, seguindo a metodologia da tecnologia escolhida, visando melhorar os aspectos de jogabilidade e interatividade.
- Analisar o desempenho do sistema, a partir de sua utilização por crianças matriculadas no Ensino Fundamental.

1.3. Estrutura da Dissertação

A dissertação está organizada em sete capítulos, incluindo esta Introdução e Conclusão.

A Introdução (Capítulo 1) apresenta a motivação para a escolha do tema, deixando explícitos o problema, o objetivo geral e as metas buscadas.

O Capítulo 2 apresenta o referencial teórico utilizado: referencial tecnológico - diz respeito ao uso de técnicas de Realidade Virtual e Sistemas Distribuídos como suporte para o desenvolvimento de jogos multi-jogadores baseados na infra-estrutura da Internet.

No Capítulo 3, são discutidos os trabalhos relacionados; é feita uma análise e comparação de entre ambientes virtuais de aprendizagem, utilizando-se o resultado de tais experiências de modo a enriquecer a elaboração do software proposto.

O Capítulo 4 apresenta características gerais do sistema protótipo. Descreve a arquitetura e funcionamento do sistema, especificando as linguagens e bibliotecas utilizadas.

O Capítulo 5 mostra detalhe da implementação do jogo e explora detalhes técnicos do seu funcionamento.

No Capítulo 6, é feita uma discussão sobre os resultados obtidos e sobre as limitações do jogo proposto.

O último capítulo (Capítulo 7) é destinado às conclusões e aponta, também, perspectivas para futuras pesquisas.

Capítulo 2

2. Realidade Virtual com Sistemas Distribuídos

2.1. Introdução

O referencial teórico da pesquisa tem como fundamento o uso de técnicas de Realidade Virtual em conjunto com Sistemas Distribuídos aplicado ao desenvolvimento de jogos multi-jogadores como base tecnológica.

Para alcançar uma melhor compreensão da “Realidade Virtual” e “Realidade Virtual Distribuída”, apresenta-se, neste capítulo, uma visão geral.

Ao final do capítulo, por meio de conceitos, componentes, requisitos e característica são apresentados com todos os parâmetros necessários para o desenvolvimento de um ambiente virtual educacional distribuído.

2.2. Realidade Virtual

A Realidade Virtual pode ser descrita como um conjunto de tecnologias que permite criar ambientes gráficos que simulam a realidade existente ou a realidade projetada (FREITAS, 2008). A RV possibilita que o usuário interaja no espaço virtual, experimentando a sensação de locomoção em três dimensões, percebendo e manipulando figuras e objetos gráficos. (CARDOSO e LAMOUNIER, 2004).

Entende-se que a interação em ambientes virtuais consiste na capacidade do usuário atuar em ambientes virtuais, promovendo alterações e reações às suas ações (KIRNER e SISCOOTTO, 2007).

Assim, verifica-se que trabalhar com um ambiente virtual permite aos usuários entender melhor os relacionamentos entre os dados e encorajar colaboração e tomada de decisões (BOTELHO, 2002). Ainda, pode-se destacar o uso da Realidade Virtual na educação

e em jogos educativos, onde busca-se explorar a integração e a interação dentro do processo de aprendizagem, despertando o senso de análise dos alunos (FREITAS *et al*, 2007).

2.2.1 Terminologia

Os termos Realidade Virtual e Ambiente Virtual são geralmente utilizados com o mesmo significado. Segundo Kirner (1996) o termo Realidade Virtual apresenta a técnica de interface, onde usuário pode realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais (KIRNER, 1996).

Apesar de Realidade Virtual e Ambientes Virtuais serem os termos mais populares alguns outros também são utilizados, tais como: Experiência Sintética; Mundos Virtuais; Mundos Artificiais e Realidade Artificial (BARNETT, 1998).

2.2.2 Definições

Como mencionado anteriormente, Realidade Virtual é uma tecnologia que vem sendo desenvolvida nos últimos anos com um grande crescimento e uma enorme expectativa. Por ser definida como uma forma avançada de interface do usuário de computador até agora disponível (BYRNE, 2005), com aplicações na maioria das áreas do conhecimento, senão em todas, há grande investimento das indústrias na produção de *hardware*, *software* e dispositivos de E/S especiais.

A Realidade Virtual pode possibilitar a criação/simulação de mundos reais ou imaginários na tela do computador, com aplicação em diversas áreas, assumindo um papel de relevo cada vez maior em campos específicos da vida econômica, social e cultural de muitos países (CAMACHO, 2005).

As técnicas de Realidade Virtual têm como objetivo imergir o usuário num mundo simulado pelo computador, por meio de estímulos sensoriais. Quanto maior o número de sentidos estimulados e melhor a qualidade destes estímulos, mais realista será a experiência. Para proporcionar este tipo de imersão são normalmente utilizados dispositivos de saída visuais, sonoros e hápticos. Entretanto, dispositivos olfativos e palatais também têm sido

experimentados, porém, estes ainda se encontram em estágio inicial de desenvolvimento (CAMELO, 2001).

Uma propriedade de grande interesse da Realidade Virtual é a sua capacidade de imersão. Isto é, sua capacidade de possibilitar a sensação de que se está fisicamente presente naquele mundo imaginário. Observam Pinho & Kirner (1998, *apud* COSTA *et al*, 2002) que é no aspecto de geração de sensações que reside o verdadeiro diferencial das interfaces de RV. Isso difere das interfaces comuns, pois o usuário de RV se sente dentro do ambiente virtual.

O nível de imersão ou de “sensação de presença” que o usuário experimenta em um ambiente virtual é determinado pelos seguintes fatores: aparência realista do ambiente virtual; resposta rápida do sistema às ações do usuário; utilização, ou não, de dispositivos atípicos de E/S, como capacetes, rastreadores de movimento, dispositivos *force-feedback* etc. (ARAUJO *et al*, 2005).

Além de imersiva, para que uma aplicação seja caracterizada como Realidade Virtual, é necessário que seja também interativa. Isto é, a Realidade Virtual não se limita apenas a uma visualização passiva porque permite ao usuário interagir com o ambiente proposto (mover objetos, por exemplo). O ambiente virtual responde em tempo real a tais ações.

Para que o usuário perceba o ambiente virtual como real, é necessário que ele possa interagir realisticamente com as entidades e objetos presentes naquele mundo. A interação do usuário com o mundo é feita principalmente por meio de sensores mecânicos e ópticos. Estes realimentam os computadores com os dados gerados a partir da captura dos gestos do usuário.

Existem diversas definições para Realidade Virtual, as quais envolvem aspectos científicos variados. Uma abordagem que apresenta uma visão geral e que sintetiza as várias discussões de RV enuncia-se a seguir: Realidade Virtual é uma interface avançada para aplicações computacionais, que permite ao usuário navegar, manipular e interagir, em tempo real, com um ambiente tridimensional gerado por computador (KIRNER e SISCOOTTO, 2007).

É importante complementar, que a interação do usuário com o ambiente virtual é um dos aspectos mais importantes da interface baseada em RV, visto que ela está relacionada com a capacidade do computador detectar e reagir às ações do usuário, promovendo alterações na aplicação (BOWMAN *et al*, 2005).

Michael Heim, em *The “Metaphysics of Virtual Reality”* (HEIN, 1994), a define como “um evento ou entidade que é real em efeito, mas não de fato”. Harrison e Jaques (1996) definem a Realidade Virtual como “a capacidade de proporcionar aos seres humanos a mais convincente ilusão de estar em outra realidade”.

Numa definição mais pragmática e também mais útil para o presente trabalho, em (CYBEREDGE, 2000) define-se a Realidade Virtual como um sistema com as seguintes características: Experiência imersiva mediada por computador; objetos virtuais modelados com o uso de técnicas de computação gráfica tridimensional; e interatividade aleatória.

2.2.3 Histórico

A realidade virtual teve suas origens na década de 60, com o desenvolvimento do *SketchPad*¹ por Ivan Sutherland (SUTHERLAND, 1963), mas só ganhou força na década de 90, quando o avanço tecnológico propiciou condições para a execução da computação gráfica interativa em tempo real.

Nos anos 60, experiências como o Sensorama (1962) e *The Sword of Damocles* (SUTHERLAND, 1968) deram os primeiros passos no sentido de um simulador multisensorial. Em seu Sensorama, Morton Heilig criou, utilizando projeção de filme colorido, som estéreo binaural, aromas e vibrações mecânicas, um ambiente virtual com todas as características necessárias para ser classificado, pelos padrões atuais, como uma aplicação em Realidade Virtual, com a exceção da interatividade.

Em 1968, foi criado o primeiro Capacete de visualização para 3D. Este capacete dispunha de dois televisores em miniatura, montados frente aos olhos. Dispunha também de um sensor de movimento.

Ivan Sutherland, com seu sistema *The Sword of Damocles* (SUTHERLAND, 1968), criou a primeira aplicação em Realidade Virtual. Ela incluía capacete 3D (*HMD – Head Mounted Display*) com rastreamento de posição e imagens 3D em tempo real (renderizada em

¹ O SketchPad foi um revolucionário programa de computador por Ivan Sutherland em 1963 durante sua tese de PhD, que usava uma interface gráfica para fazer desenhos em 2D interativamente com o usuário.

aramado, ou wireframe). Este projeto lançou as fundações para o surgimento da Realidade Virtual como a conhecemos hoje.

Na década de 70, o projeto GROPE, da Universidade da Carolina do Norte, desenvolveu os primeiros protótipos de dispositivos de *force-feedback*² (BROOKS, 1990). No mesmo período, Myron Krueger desenvolveu o *VideoPlace*, um ambiente conceitual sem existência. Nele, as silhuetas dos usuários eram captadas por câmeras de vídeo e projetadas numa grande tela. Por meio de técnicas de processamento de imagens, que podiam determinar o posicionamento do usuário na tela, o equipamento permitia que usuários interagissem com o ambiente gerado pelo computador (JACOBSON, 1994).

A década de 80 foi crucial para o desenvolvimento da Realidade Virtual. Foram realizados vários estudos e projetos, envolvendo hardware e sensoriamento remoto. A área de Realidade Virtual, assim como outras áreas só se desenvolveram com a convergência de uma série de fatores, incluindo-se as pesquisas, a tecnologia, a disponibilidade de produtos e os custos acessíveis (KIRNER, 2006).

Nos anos 80, Jaron Lanier (2000) cunhou o termo Realidade Virtual. Lanier desenvolveu um dos primeiros dispositivos de interface imersivos (LANIER, 2000). Apesar de o termo ter sido preterido por muitos pesquisadores, ele continua sendo o mais comumente utilizado. Foi durante esta década que a Realidade Virtual realmente despontou como uma nova área da disciplina de interfaces computacionais. A seguir, enumeramos alguns dos avanços da época:

- **VCASS** (*Visually Coupled Airborne Systems Simulator*): foi desenvolvido por Thomas Furness nos Laboratórios de Pesquisa Médica da Força Aérea Americana em 1982. Tratava-se de um simulador no qual o piloto utilizava um HMD que aumentava a visão externa do usuário, fornecendo-lhe gráficos com informações sobre alvos e trajetórias de vôo (FURNESS, 1968).
- **BOOM**: Comercializado a partir de 1989, pela Fake Space Labs. Tratava-se de uma pequena caixa contendo dois monitores CRT que podiam ser vistos por meio de orifícios num de seus lados. O usuário segurava a caixa com as mãos,

² Também chamados de haptic interfaces os sistemas que produzem sensação de tato (touch feedback) ou de força (force feedback) são usados em ambientes de Realidade Virtual para aumentar a sensação de imersão

mantendo-a próxima aos olhos enquanto a movia pelo mundo virtual. O braço mecânico, ligado à caixa, media a posição e a orientação da mesma (TURUNEM, 2000).

- **UNC Walkthrough Project:** Com o objetivo de melhorar a qualidade dos HMDs, rastreadores posicionais ópticos e dispositivos de tela, a Universidade da Carolina do Norte desenvolveu, em meados da década de 80, uma aplicação arquitetônica com a qual o usuário podia passear por edificações virtuais (BROOKS, 1992).
- **VIVED:** Em 1984, o *Ames Research Center*, da NASA, desenvolveu equipamentos de Realidade Virtual de baixo custo, utilizando TVs com telas de cristal líquido e outros componentes de consumo de massa. Isto foi feito com o objetivo de baixar o custo exorbitante dos equipamentos utilizados para a construção de simuladores de voo (JACOBSON, 1994).
- **Data-Glove:** Em 1985, a empresa VLP começou a comercializar os primeiros dispositivos de RV do mercado. O primeiro deles foi o Data-Glove que conseguia ler a posição da mão e dos dedos do usuário, permitindo que ele manipulasse objetos e interagisse com o mundo virtual por meio de gestos (JACOBSON, 1994).

Foi na década de 90, porém, que a RV deixou de ser apenas um brinquedo extravagante. Diversas aplicações comerciais, educacionais e médicas passaram a utilizar a RV como principal meio de interação. A CAVE e a *ImmersaDesk* são duas das tecnologias desenvolvidas na referida década (CRUZ-NEIRA, 1992).

A primeiro delas, CAVE (*Automatic Virtual Environment*), é composta por uma pequena sala em que, no chão e nas paredes, são projetadas imagens estereoscópicas do mundo virtual. O usuário, que utiliza óculos obturadores de cristal líquido para tirar proveito do efeito de profundidade, fica totalmente imerso nas imagens de alta resolução (Electronic Visualization Laboratory, 2000).



Figura 2-1: Desenho Esquemático de uma CAVE.

Fonte: Electronic Visualization Laboratory

Já a *ImmersaDesk* é um dispositivo semi-imersivo com o formato de uma prancheta de desenho. O usuário, que utiliza óculos estéreos, fones de ouvido e dispositivo de rastreamento óptico, pode observar uma imagem projetada de alta resolução, com campo de visão de até 90°.



Figura 2-2 : Desenho Esquemático de uma ImmersaDesk.

Fonte: Electronic Visualization Laboratory.

2.2.4 Os diferentes níveis de experimentação

Num ambiente virtual, todas as impressões sensoriais são geradas por um computador. O tipo e a qualidade destas impressões determinam o nível de imersão e a sensação de presença. Perante o usuário, um ambiente virtual deve: ter uma aparência realística; e reagir realisticamente às suas ações.

Os atuais sistemas de RV são capazes de satisfazer apenas parcialmente estas expectativas. Luis Brill, em (BRILL, 1994) define os sete seguintes tipos de Realidade Virtual:

- **Janela para o Mundo:** Este é o tipo mais simples de Realidade Virtual. Utiliza um monitor convencional para exibir imagens do ambiente virtual. Por sua simplicidade, praticidade e baixo custo é o tipo mais comumente utilizado. Exemplos de aplicações incluem o jogo Quake – Arena, da Id Software (IDSOFTWARE, 2000) e o visualizador de VRML Cortona, da Parallelgraphics (PARALLELGRAPHICS, 2008).
- **Imersiva, em Primeira Pessoa:** Fornece uma experiência imersiva, em primeira pessoa, na qual o usuário é posto dentro do ambiente virtual. Geralmente, é feita com a utilização de *HMDs* e *Data-Gloves*³. Exemplos de aplicações incluem o jogo Magic Carpet Ride, da Walt Disney e o visualizador de campos Virtual Wind Tunnel, da NASA (PAUSCH, 1996) (NASA, 2000).
- **Mundo Espelhado:** Promove uma experiência em terceira pessoa, com o usuário vendo sua própria imagem sobreposta ou composta com o ambiente virtual. Geralmente, utiliza grandes monitores ou telas de projeção. Alguns exemplos são o *VideoPlace*, de Myron Krueger e o *ImmersaWall*, da Universidade da Carolina do Norte (KRUEGER, 2008).
- **Mundo de Waldo:** Equivalente em Realidade Virtual ao teatro de fantoches. Utiliza sensores mecânicos para capturar gestos e expressões de atores humanos e computadores para reproduzir, por meio de técnicas de Computação Gráfica, estes movimentos e expressões em bonecos virtuais. Exemplos incluem Virtual Actors, da SimGraphics Engineering e Flock of Birds, da Ascension (SIMGRAPHICS, 2000).
- **Mundo de Câmara:** Teatro de projeção em RV. Fornece aos usuários uma grande sensação de imersão, porém, com baixo nível de interatividade. Sons e

³ A Data-Glove é uma luva conectada ao computador equipada com sensores detectando a posição da mão, pois calculando ângulos e grau das movimentações.

cheiros podem ser incorporados. Um exemplo de aplicação é o Teatro em RV, da Embraer (CAMELO, 2001).

- **Simulação em Cabine:** Experiência de RV em primeira pessoa. O usuário permanece numa cabine e pode observar o mundo virtual por meio de suas janelas. Exemplos são os simuladores de vôo militares e comerciais e o jogo *Battle Tech*, da *Virtual Worlds Entertainment*.

2.2.5 Imersão, interação e envolvimento

A RV também pode ser caracterizada pela coexistência integrada de três idéias básicas: imersão, interação e envolvimento (MORIE, 1994). A idéia de imersão está ligada ao sentimento de fazer parte do ambiente. Normalmente, um sistema imersivo é obtido com o uso de capacete de visualização, ou cavernas; sistemas imersivos baseados em salas com projeções das visões nas paredes, teto, e piso (CRUZ-NEIRA, 1992). Além do fator visual, dispositivos ligados aos demais sentidos também são importantes para o sentimento de imersão, como o som (BEGAULT, 1994; GRADECKI, 1994), o posicionamento automático da pessoa e dos movimentos da cabeça, controles reativos, etc. A visualização de uma cena 3D em um monitor é considerada não imersiva. Dessa forma, tem-se a conceituação de RV imersiva e não imersiva (LESTON, 1996). De modo geral, do ponto de vista da visualização a RV imersiva utiliza capacete ou cavernas, enquanto a RV não imersiva utiliza monitores. Entretanto, dispositivos baseados nos demais sentidos podem introduzir algum grau de imersão à RV que usa monitores (ROBERTSON, 1993). Os monitores ainda apresentam alguns pontos positivos, como o baixo custo e a facilidade de uso, evitando as limitações técnicas e problemas decorrentes do uso do capacete. Porém, a tendência deve ser a utilização da RV imersiva na grande maioria das aplicações futuras.

A interação está ligada à capacidade do computador detectar as entradas do usuário e modificar instantaneamente o mundo virtual em função das ações efetuadas sobre ele (capacidade reativa). As pessoas são cativadas por uma boa simulação em que as cenas mudam em resposta aos seus comandos, que é característica mais marcante dos vídeo games. Para que um sistema de RV pareça mais realista, o ambiente virtual inclui objetos simulados. Outros, artifícios para aumentar o realismo são empregados, por exemplo, a texturização dos

objetos do ambiente e a inserção de sons tanto ambientais quanto sons associados a objetos específicos (ARAÚJO, 1996).

A idéia de envolvimento, por sua vez, está ligada ao grau de motivação para o engajamento de uma pessoa em determinada atividade. O envolvimento pode ser passivo, como ler um livro ou assistir televisão, ou ativo, ao participar de um jogo com algum parceiro. A RV tem potencial para os dois tipos de envolvimento ao permitir a exploração de um ambiente virtual e propiciar a interação do usuário com o mundo virtual dinâmico.

Embora a percepção visual seja nosso sentido primário, outros sentidos também devem ser estimulados para proporcionar uma completa imersão; entre os quais o retorno auditivo, o tato e a força de reação.

2.2.6 RV passiva, exploratória ou interativa

Um aplicativo de RV pode proporcionar uma sessão sob três formas diferentes: Passiva, Exploratória ou Interativa (ADAMS, 1994). Uma sessão de RV passiva proporciona ao usuário uma exploração do ambiente automática e sem interferência. A rota e os pontos de observação são explícitos e controlados exclusivamente pelo software. O usuário não tem controle algum, exceto talvez, para sair da sessão.

Uma sessão de RV exploratória proporciona uma exploração do ambiente dirigida pelo usuário. O participante pode escolher a rota e os pontos de observação, mas não pode interagir de outra forma com entidades contidas na cena (ADAMS, 1994).

Uma sessão de RV interativa proporciona uma exploração do ambiente dirigida pelo usuário e, além disso, as entidades virtuais do ambiente respondem e reagem às ações do participante. Por exemplo, se o usuário move o ponto de observação em direção à porta, esta pode parecer abrir-se, permitindo ao participante passar por ela.

A navegação em ambientes virtuais é controlada pelo posicionamento do observador no mundo virtual, o chamado ponto de observação (viewpoint). A navegação é, provavelmente, a forma mais simples de interação encontrada em muitas aplicações de RV. Virtualmente, toda técnica de navegação pode ser deduzida a partir de um único modelo, que

assume uma câmera montada em um “carro” virtual que, às vezes, é referenciado como Flying Carpet model. Algumas das técnicas de navegação mais usuais são:

- ***Point-and-fly***: o usuário move o carro virtual apontando em uma determinada direção com os dispositivos de navegação (por exemplo, uma luva), e faz um certo gesto ou aperta determinado botão. Se a luva estiver sendo usada, a velocidade de movimentação pode ser controlada. Se o rastreador de cabeça estiver ativo, o mesmo controlará a câmera virtual. Essa técnica é uma das mais usadas e sugere um modo mais sofisticado: o usuário aponta para um objeto desejado e o sistema computa o desvio que posicionará o usuário em frente ao objeto; aqui também a velocidade pode ser controlada. Algumas vezes, é desejável restringir o carro a uma certa altura, por exemplo, ao nível dos olhos, e o usuário pode então se mover ao redor da cena, nessa altura fixa;
- ***Eyeball-in-hand***: esse paradigma é implementado pela realimentação do sistema de rastreamento diretamente para o ponto e a direção de observação, enquanto o ponto de observação permanece fixo. Essa técnica é muito apropriada para a análise de um único objeto a partir de diferentes pontos de observação, por exemplo, no projeto de interiores. Em resumo, todas as ações são carregadas/atualizadas a partir do ponto de observação do usuário (TURUNEM, 2002).
- ***Scene-in-hand***: este paradigma é oposto ao anterior (TURUNEM, 2002). A técnica mapeia os movimentos dos dispositivos de entrada 3D dentro de um mundo virtual, mantendo o ponto de observação do usuário e movendo o mundo em volta. Um mecanismo de “ação das mãos” (pegar, soltar, etc.) é, normalmente, incluído para ampliar o alcance dos movimentos (HAND, 1997).
- Algumas vezes é necessário controlar o ponto de observação sem o uso das mãos. Nesse caso, o reconhecimento de voz pode ser empregado para mover o carro virtual, proferindo comandos simples como: à esquerda, pare, etc.

Para aumentar a flexibilidade é altamente desejável que diferentes modos de navegação sejam mapeados para diferentes configurações de dispositivos de entrada, já que certas combinações de modos de navegação e configurações de dispositivos de entrada tendem a ser usadas com mais frequência do que outras. Existem muitos parâmetros que

afetam a representação do usuário e a navegação, como velocidade de navegação, tamanho da mão, escala dos movimentos da cabeça (head motion) e separação dos olhos.

2.2.7 Aplicações

Muitas são as aplicações da RV. Com utilização crescente nas mais variadas áreas, desde jogos de videogames até cirurgias a distância, a RV vem se tornando cada vez mais comum. Apesar disto, para que haja uma adoção mais ampla, ainda é necessário que aconteça uma melhoria na acessibilidade dos equipamentos, tanto em termos de custo quanto em usabilidade. A seguir listaremos alguns dos usos mais frequentes desta tecnologia:

- **Treinamento:** A RV equipa hoje todos os simuladores de vôo utilizados para o treinamento de pilotos civis e militares. Ela também é utilizada para o treinamento tático e operacional de soldados. A indústria também aplica técnicas de Realidade Virtual para treinar funcionários na operação de maquinário sensível e em como agir em situações de emergência (SINGHAL apud ZYDA, 1999).
- **Manufatura:** Indústrias como a da aviação e a automobilística começam a aplicar a Realidade Virtual para produtos ainda em fase inicial de projeto. Por meio da Realidade Virtual é possível avaliar, principalmente, atributos ergonômicos, tais como acessibilidade de peças durante manutenção, posicionamento de comandos e controles, visibilidade (DIAS, 2006).
- **Entretenimento:** A indústria de entretenimento é um dos principais motores do desenvolvimento da Realidade Virtual. Com os mais variados graus de imersividade, diferentes aplicações existem na área: jogos em primeira pessoa em consoles de videogames e computadores pessoais, passeios de imersão parcial em salas de projeção, jogos multi-jogadores em imersão total distribuído em redes entre várias máquinas, entre outras (KOZOVITS, 2003).
- **Medicina:** A Realidade Virtual é utilizada na medicina para a realização de cirurgias a distância. Neste tipo de aplicação, o médico tem condições de operar um paciente a milhares de quilômetros de distância, utilizando luvas

interativas e óculos 3D. Na outra ponta, o paciente é operado por um robô que reproduz os movimentos realizados pelo médico (BRAS, 2001).

- **Arquitetura:** Utilizando-se técnicas de Realidade Virtual, é possível caminhar por dentro de construções que só existem no computador. Assim, pode-se identificar e corrigir defeitos antes mesmo que uma obra seja iniciada, a um custo muito pequeno. A Realidade Virtual também pode ser utilizada como ferramenta de vendas, permitindo que potenciais compradores visitem, por exemplo, apartamentos decorados de prédios ainda em construção (RIBEIRO, 2006).
- **Educação:** Principalmente por seu custo, a aplicação da Realidade Virtual na educação ainda está restrita a laboratórios de universidades e centros de pesquisa. As possibilidades de aplicação da tecnologia na educação são ilimitadas e vão desde a visualização de fenômenos naturais invisíveis ao homem até a observação de lugares ou eventos históricos, passando por sua utilização em simuladores, na educação a distância e na formação de comunidades virtuais. O desenvolvimento do potencial das tecnologias digitais a serviço do ensino é, por outro lado, uma das aspirações de inúmeros programas de fomento à pesquisa e de instituições governamentais voltadas à educação. Citam-se entre esses inúmeros programas o projeto EDUCOM, realizado em cinco universidades: Universidade Federal de Pernambuco (UFPE), Universidade Federal de Minas Gerais (UFMG), Universidade Federal do Rio de Janeiro (UFRJ), Universidade Federal do Rio Grande do Sul (UFRGS) e Universidade Estadual de Campinas (UNICAMP). Outros programas voltados a projetos de informática na educação são ainda a Rede Interativa de Educação (RIVED) e o Programa Nacional de Informática na Educação (PROINFO) (FREITAS, 2008).

2.3. Realidade Virtual Distribuída

Os termos “Ambiente Virtual” e “Ambiente Virtual Distribuído” são recentes e permanecem sem uma definição única que os caracterizem. Desta forma, para alcançar uma melhor compreensão, apresenta-se neste capítulo, uma visão geral e os objetos de estudo deste

trabalho. O uso intensivo de tecnologia vem sofrendo uma incrível expansão. Significa a possibilidade de criar ambientes que proporcionem melhores condições para a aquisição e construção do conhecimento.

2.3.1 Conceituação e Caracterização

Ambientes virtuais distribuídos figuram entre os sistemas de software mais complexos já construídos (STY TZ, 1996). Esses ambientes devem satisfazer uma variedade de características tais como:

- (1) Resposta rápida a novos requisitos do sistema.
- (2) Capacidade de manutenção.
- (3) Suporte para interação em tempo real.
- (4) Fidelidade da inserção do usuário no AV em relação a uma referência.
- (5) Alta taxa de quadros por segundo, reusabilidade e portabilidade.
- (6) Ajustamento a novas interfaces e dispositivos de visualização.
- (7) Requisitos para capacidades adicionais.

A elaboração de um sistema de Realidade Virtual distribuído inclui atividades envolvendo (STY TZ, 1996):

- (1) Suporte de comunicação em rede.
- (2) Criação de ambientes virtuais.
- (3) Atuação no mundo real.
- (4) Criação de atores gerados por computador.
- (5) Inserção de fenômenos naturais.
- (6) Uso de simulação tradicional.

O suporte de comunicação em rede fornece os meios para que as unidades computacionais heterogêneas, separadas fisicamente, sejam unificadas para programar um único ambiente virtual.

Devido à necessidade de comunicação intensa entre os usuários do sistema, foi desenvolvida a técnica de *dead-reckoning* (STY TZ, 1996), pois não é usada em todos os casos, mas trata de minimizar a troca de mensagens e suportar atrasos de comunicação. Essa abordagem trabalha com a previsão da posição de um elemento, levando em conta o seu trajeto, velocidade e posição anterior, decorrido certo tempo. Todas as máquinas fazem o mesmo cálculo de previsão e reposicionam o elemento. Aquele que estiver gerenciando o elemento conseguirá verificar a diferença da trajetória real com a trajetória calculada. Sempre que essa diferença atingir um valor máximo, o valor real da posição será então comunicado às outras máquinas para devida correção. Desta forma, não haverá necessidade de informar continuamente a posição de um elemento para as outras máquinas, diminuindo bastante a comunicação pela rede.

Os sistemas de RV multi-usuários em ambiente distribuído vêm crescendo e apresentam elevado potencial de aplicações. Este tipo de sistema permite que os usuários geograficamente dispersos atuem em ambientes virtuais compartilhados, usando a rede para melhorar o desempenho coletivo, por meio da troca de informações (ARAÚJO, 1996).

Um Ambiente Virtual Distribuído ou AVD pode ser definido como um sistema por meio do qual, diversos usuários interage entre si e em tempo real, podendo os mesmos estar situados em localidades diferentes. Tipicamente, cada usuário acessa seu próprio computador, usando-o para fornecer uma interface para o ambiente virtual. Estes ambientes geralmente oferecem aos usuários uma sensação de realismo por meio da incorporação de gráficos 3D e som estéreo, a fim de criar uma experiência imersiva (SINGHAL *apud* ZYDA 1999).

De acordo com Sementille (1999), um AVD pode ser evidenciado por cinco características comuns. Tais características são: Sensação de Compartilhamento de Espaço, Sensação de Presença, Sensação de Tempo Compartilhado, Comunicação entre os Participantes, Forma de Compartilhamento. A seguir será feita uma breve descrição de todas essas características.

1. **Sensação de Compartilhamento de Espaço:** todos os participantes têm a ilusão de estarem localizados no mesmo lugar, tais como na mesma sala, prédio ou

região. Este espaço compartilhado representa um local comum, no qual outras interações podem acontecer. Este local pode ser real ou fictício. O local compartilhado deve apresentar as mesmas características a todos os participantes.

2. **Sensação de Presença:** quando entra em um local compartilhado, cada participante torna-se uma “pessoa virtual”, chamada avatar, o qual inclui uma representação gráfica, um modelo de estrutura corporal (por exemplo, a presença de braços, cabeça, juntas e etc), um modelo físico (por exemplo, peso altura, etc.), e outras características. Uma vez dentro de um AVD, cada participante pode ver os outros avatares localizados no mesmo espaço. Similarmente, quando um usuário deixa o AVD, os outros participantes vêem seu avatar partir. Nem todos os participantes precisam ser controlados por seres humanos. Alguns participantes podem ser entidades sintéticas controladas por modelos de simulação dirigidos por eventos.

3. **Sensação de Tempo Compartilhado:** os participantes devem ser capazes de ver o comportamento uns dos outros em tempo real.

4. **Comunicação entre os Participantes:** embora a visualização seja a base para um AVD efetivo, muitos AVDs também empenham-se em permitir que algum tipo de comunicação ocorra entre os participantes a qual pode acontecer por meio de gestos, escrita ou voz. Esta comunicação adiciona uma sensação maior de realismo a qualquer ambiente simulado e é fundamental para sistemas de engenharia e treinamento.

5. **Forma de Compartilhamento:** os elementos mencionados anteriormente efetivamente oferecem um sistema de videoconferência de alta qualidade. O poder real de um AVD deriva da habilidade dos usuários em interagir uns com os outros e com o próprio ambiente virtual. Em uma simulação de combate ou jogo, por exemplo, os usuários precisam atirar ou colidir uns com os outros. Estas interações precisam ser modeladas realisticamente. Os usuários podem ser capazes de escolher, mover, e manipular objetos que existem no ambiente e até entregá-los a outros participantes.

A importância dos AVDs reside na integração dos gráficos imersivos com a habilidade de possuir diversos usuários compartilhando informação e manipulando objetos no ambiente. A presença de diversos usuários independentes diferencia os AVDs dos sistemas de Realidade Virtual tradicionais. Os AVDs são mais apropriados para aplicações que demandam a criação de tele-presença, ou seja, a ilusão que outros usuários estão visíveis, mesmo que situados

fisicamente em lugares remotos. Nestas aplicações, os usuários sentem uma sensação de realismo próxima à alcançada em um contato face a face (RIBEIRO, 2006, p.50-52).

2.3.2 Os componentes da Realidade Virtual Distribuída

Um sistema de Realidade Virtual Distribuída consiste de quatro componentes básicos: 1) display gráfico; 2) dispositivos de comunicação e controle; 3) um sistema de processamento; 4) e uma rede de comunicação. Estes componentes trabalham juntos para fornecer a sensação de imersão em diferentes localidades (RIBEIRO, 2006), (SEMENTILLE, 1999), (SINGHAL apud ZYDA 1999).

Além desses componentes, aspectos quanto ao desenvolvimento de software para AVDs, também são muito importantes e envolvem a comunicação em redes, visões do ambiente, modelo de dados, gerenciamento da computação, comportamento dos objetos e técnicas de distribuição (MACEDONIA, 2005).

2.3.3 Comunicação em Rede

Diversas características da comunicação em rede devem ser consideradas e suas principais características são: a largura de banda, a latência, a confiabilidade e os esquemas de comunicação (VLOKA, 1995).

2.3.3.1 Largura de Banda

A largura de banda é a taxa na qual a rede pode entregar um dado a um computador (host) destinatário. É a quantidade de informação que pode ser transferida de um nó para outro em um determinado período. Algumas aplicações têm de transmitir dados, a certa velocidade para serem efetivas. Por exemplo, se uma aplicação de telefonia por Internet codifica a voz a 32 Kbps, então ela deve poder também enviar os dados para a rede e fazer com que sejam entregues na aplicação receptora a essa mesma taxa. Se essa largura de banda não estiver disponível, a aplicação precisará codificar a uma taxa diferente, ou então, preterir-la, pois receber metade da largura de banda de que precisa de nada adianta para tal aplicação sensível à largura de banda (VLOKA, 1995; KUROSE, 2003).

2.3.3.2 Latência

A latência de uma rede é o tempo requerido pela rede para transferir um bit de dado de um ponto a outro. A latência de rede pode variar muito. Se o dado for enviado por meio de uma rede local, a latência tipicamente gira em torno de poucos milisegundos (10 a 20 ms). Porém, se a comunicação for feita usando-se a Internet, por meio de um modem e do sistema telefônico, sua latência será de, no mínimo, 100 milisegundos (SINGHAL *apud* ZYDA, 1999).

2.3.3.3 Confiabilidade de Rede

A confiabilidade da rede é uma medida que reflete a quantidade de dados que são perdidos pela rede durante o trajeto do host fonte para o destino. O dado perdido pode ser dividido em duas categorias: dado interrompido e dado corrompido. Muitos pesquisadores consideram a interrupção e a corrupção de dado essencialmente equivalente, isto porque, em ambos os casos, o dado não atinge efetivamente a aplicação destino (SINGHAL *apud* ZYDA 1999).

2.3.3.4 Esquema de Comunicação

O modelo de comunicação tem muita influência com a escalabilidade de um sistema. A escalabilidade pode ser mensurada pela quantidade de entidades que podem, simultaneamente, participar do sistema (SEMENTILLE, 1999). Alguns esquemas escalam melhor do que outros. Entre os principais modelos de comunicação tem-se:

- *broadcast* (por difusão - ocorre quando o dado é enviado para todos os hosts, ou seja, uma mensagem enviada por um host é recebida por todos os outros que podem até ignorá-la; este modelo possui alto custo e é possível somente em redes locais);
- *unicast* (ponto a ponto - ocorre quando se estabelece uma comunicação entre dois *hosts*);
- *multicast* (ocorre quando um grupo de *hosts* se comunica pela rede, via uma única transmissão) (KIRNER, 1988).

2.3.4 Visões do Usuário de um AVD

A visão (ou perspectiva) que um usuário ou processo tem do ambiente virtual é geralmente representada por meio de janelas. Existem dois paradigmas principais para visão em ambientes distribuídos: a visão síncrona e a visão assíncrona.

- **Visão síncrona:** nesse tipo de paradigma têm-se várias máquinas controlando visões diferentes de um mesmo ambiente. As imagens exibidas por estas máquinas devem ser sincronizadas de forma a dar a ilusão de conjunto. Como exemplo, um simulador de voo, em que máquinas diferentes controlam a visão da frente, da direita, da esquerda e da cabine. As imagens são coordenadas de maneira a parecer que todas fazem parte de uma única visão da cabine. Alguns AVDs que usam visão síncrona são: RAVEN (CATER, 1995) e o CAVE. Por exigir alta confiabilidade e menor latência, a visão síncrona é restrita a redes locais.
- **Visão assíncrona:** esse paradigma é mais geral e permite que múltiplos usuários tenham controle individual sobre "quando" e "o que" pode estar fisicamente separado por uma rede local ou uma rede a longa distância. Como exemplo de AVDs que utilizam a visão assíncrona tem-se o NPSNET (MACEDONIA *et al*, 1994).

2.3.5 Modelo de Dados

A determinação do modelo de dados a ser utilizado em um ambiente virtual, talvez, a decisão de construção mais difícil de ser tomada, pois afeta o escalonamento, os requisitos de comunicação.

Existem diversos conceitos para a distribuição persistente ou semi-persistente de dados. As consideradas mais relevantes são:

- **Modelo da Base de Dados Homogênea Replicada:** nesse modelo, tem-se uma base de dados homogênea que contém todas as informações sobre a geometria, texturas e comportamento de todos os objetos que integram o ambiente virtual. Esta base de dados é, então, replicada em todos os hosts que participam do AVD. Uma vantagem desse modelo é que as mensagens são geralmente pequenas, uma vez que os participantes só são informados sobre as

mudanças de estado dos objetos, como, por exemplo, mudanças de posição e colisões entre objetos. As desvantagens são que a base de dados de usuário tende a crescer com o aumento do conteúdo do ambiente virtual (RIBEIRO, 2006, p.33).

- **Modelo da Base de Dados Centralizada e Compartilhada:** neste modelo, as máquinas dos participantes comunicam-se com um único servidor central, o qual mantém a base de dados com todas as informações e estado do mundo virtual. A principal vantagem desse modelo reside na facilidade de manutenção da consistência da base de dados. A principal desvantagem é que o uso de um servidor centralizado para mundos virtuais 3D limita o número de participantes devido à concentração de E/S e a manutenção de uma base de dados de objetos, em tempo real. O servidor central torna-se rapidamente um "gargalo" do sistema, já que deve retransmitir o estado de cada participante a todos os demais (RIBEIRO, 2006, p.33).
- **Modelo da Base de Dados Distribuída, Compartilhada, Ponto a Ponto:** a idéia desse modelo é fazer com que o sistema distribuído simule as arquiteturas de memória compartilhada. A base de dados é homogênea, distribuída e completamente replicada entre as máquinas dos participantes. Como exemplo, tem-se o sistema DIVE (CARLSSON, 1993), no qual a base de dados inteira é dinâmica e utiliza protocolos *multicast* confiáveis para replicar novos objetos. A desvantagem se verifica em relação aos custos de comunicação, associados com a manutenção da confiabilidade e da consistência dos dados (RIBEIRO, 2006, p.34).
- **Modelo da Base de Dados Distribuída, Compartilhada, Cliente/Servidor:** outro modelo que pode ser utilizado é uma variação do modelo cliente/servidor, no qual a base de dados é particionada entre os clientes e a comunicação é mediada por um servidor central. Nesse sistema, o servidor central possui o conhecimento de qual parte do mundo virtual reside em cada cliente. Toda vez que um objeto se move no ambiente virtual, a base de dados é atualizada por intermédio do servidor. Todavia, em um ambiente virtual dinâmico de grande escala, os servidores tendem a se tornar os "gargalos" de

entrada e saída do sistema, aumentando latência inerente ao ambiente (RIBEIRO, 2006, p.34).

2.3.6 Gerenciamento da Computação

A computação, tanto quanto os dados, também pode ser distribuída. Como computação, no caso, entende-se qualquer tarefa, envolvendo um objeto específico. Os modelos de gerenciamento mais importantes são:

- **Distribuição Completa:** nesse modelo, todas as operações sobre um objeto devem ser executadas no mesmo nó que mantém os dados sobre aquele objeto. Se um objeto deseja executar uma operação sobre outro, deve, então, enviar uma mensagem para o outro processo. A alocação de processos aos nós pode ser otimizada por meio da utilização de algoritmos para balanceamento de carga. Monitorando-se o consumo de recursos e padrões de comunicação, pode-se derivar uma alocação ótima. Isto permitirá que muitos objetos, que se comunicam com frequência entre si, possam ser colocados no mesmo nó. Esta movimentação de processos é comumente conhecida como migração. Essa abordagem funciona bem em fortemente acoplados (RIBEIRO, 2006, p.34).
- **Distribuição Parcial:** este método é similar ao da distribuição completa, exceto pelo estado dos objetos que é normalmente adquirido, usando-se técnicas de replicação parcial de dados e as mudanças são feitas localmente. Porém, não existe duplicação de esforço computacional (RIBEIRO, 2006, p.35).
- **Replicação Parcial:** para compensar ligações de comunicações mais lentas, por exemplo, como nos sistemas fracamente acoplados, é possível replicar o estado da computação em alguns ou todos os nós. Estes processos-réplica são geralmente usados para aproximar o comportamento do objeto, usando-se o método conhecido como *dead-reckoning*. O processo, que executa a completa simulação daquele objeto, também processa esse modelo em paralelo e, quando há dois diferentes de uma quantidade pré-definida, uma cópia das variáveis de estados reais daquele objeto é enviada para todos os processos-réplica. As

subseqüentes aproximações são, então, baseadas na última atualização (RIBEIRO, 2006, p.35).

- **Replicação Total:** a simulação de cada objeto em cada nó pode ser requerida, se a simulação está sendo executada numa rede de longa distância. Neste modelo, somente a informação, que muda o comportamento dos objetos espelhados, é enviada, fazendo com que todos os cálculos sejam feitos localmente. Por conseguinte, o comportamento parece correto em todo lugar (apesar de poder não ser exatamente ao mesmo tempo), mas ao custo da duplicação de computação (RIBEIRO, 2006, p.35).

2.3.7 Comportamento dos Objetos

Atualmente, ainda é tema de discussão sobre o que consiste o "comportamento" do objeto e como este deve ser modelado. No estrito senso da orientação a objeto, os dados são atributos e os métodos manipulam os atributos de uma forma pré-definida, por exemplo, mudando de posição a todo instante. Portanto, combinando-se dados e métodos, consegue-se a impressão de "comportamento". No entanto, a carga computacional requerida para suportar este comportamento do objeto pode ser muito alta.

É possível classificar o "comportamento" do objeto como determinístico ou não-determinístico. Em geral, objetos que não necessitam de entradas (via dispositivos de entrada) são determinísticos, ao passo que aqueles objetos que necessitam de entradas, inclusive aqueles sob o controle de humanos, são não-determinísticos. Por exemplo, as decisões tomadas por um carro-robô podem ser determinadas durante seu avanço, porém, em um carro virtual sendo dirigido por um humano, em um simulador de direção, isto não pode ser previsto (RAWKES, 1996). A capacidade de prever o comportamento significa que é possível melhorar a comunicação e a latência do sistema.

Segundo Roehl (2004), o comportamento determinístico pode ser subdividido em duas subcategorias: estática e animada. Similarmente, o comportamento não-determinístico pode ser Newtoniano ou inteligente. O estado de um objeto estático é constante e, portanto, 100% previsível, a todo o momento; já um objeto animado muda seu estado a todo instante, mas ainda é previsível. Um objeto Newtoniano interage com seu ambiente, mas não de uma

maneira direta. Portanto, um objeto inteligente pode ter um comportamento complexo e tão imprevisível quanto o de um ser humano.

2.3.8 Técnicas da Realidade Virtual Distribuída

A Realidade Virtual Distribuída enfrenta o problema do gerenciamento do estado compartilhado. Entretanto, existem técnicas para controlar e melhorar o tempo para uma mensagem chegar ao seu destino. Neste trabalho adota-se algumas destas “Técnicas” mencionadas na implementação do estudo de caso proposto.

a) Dead-Reckoning: teve sua origem em simulações militares, nas quais se tentava prever as trajetórias de aviões e veículos. A técnica tenta estimar a posição de um objeto baseado em alguma informação do passado sobre esse objeto (SINGHAL, 1999 *apud* ZYDA, 1999). Por exemplo, considere que, no passado, um objeto estava na posição (0,0), movendo a 10m/s na direção (1,0). Após 2 segundos, pode-se estimar que a posição do objeto seja representada pelo seguinte polinômio:

$$\text{posição.x} ::= \text{posição.x} + (\text{velocidade} * \text{direção.x}) * \text{variação (Seg)};$$

$$\text{posição.y} ::= \text{posição.y} + (\text{velocidade} * \text{direção.y}) * \text{variação (Seg)}.$$

Com a finalidade de reduzir o número de mensagens enviadas entre clientes, enquanto um cliente souber que os outros clientes são capazes de calcular a posição de um objeto sob seu controle, não há necessidade de enviar uma nova mensagem sobre o objeto. Só quando o cliente faz alguma alteração no objeto, torna-se impossível aos outros clientes estimarem corretamente a sua posição, havendo, então, necessidade do envio de uma nova mensagem sobre o objeto. Ao receber uma mensagem de atualização, o objeto remoto recupera a consistência de estado, convergindo para o estado físico transmitido. Há problemas de descontinuidade que podem ser razoavelmente solucionadas com ajuste de curvas (SINGHAL 1999 *apud* ZYDA, 1999) e protocolos baseados na história das posições (SINGHAL *et al*, 1994). Estas soluções, entretanto, não conseguem evitar colisões indesejáveis que ocorrem quando a trajetória prevista difere da real.

Uma maneira de se reduzir ainda mais o número de mensagens trocadas e evitar colisões indesejáveis, é por meio de um *Dead-Reckoning* baseado em objetivos

(SZWARCMAN *et al*, 2001). Para esse *Dead-Reckoning* funcionar é necessário que os clientes e servidores possuam inteligência que os tornem capazes de decidir sobre a movimentação de um objeto baseado em seu objetivo. Esses objetivos podem ser aleatórios, tais como atacar um determinado alvo.

Nessa forma de *Dead-Reckoning*, os objetivos são enviados pela rede. Quando um cliente recebe um novo objetivo de um determinado objeto, ele passa a calcular a movimentação de maneira que o objeto alcance o objetivo desejado. Isso pode causar variações significativas na movimentação de um mesmo objeto entre clientes e pode não ser uma boa solução em todos os jogos em que o *Dead-Reckoning* é aplicável.

b) Sincronização baseada em tempo de comando: O jogador clica na peça e movimenta para um destino, essa peça começa a se mover para a posição desejada. Para que os outros jogadores vejam o mesmo movimento, uma mensagem é enviada para o servidor pelo cliente que requisitou o movimento e depois o servidor reenvia a mensagem de movimento para os outros clientes. Um exemplo disso é mostrado a seguir:

Um cliente move ao destino (30,0) fazendo uma peça na posição (0,0) começar a se mover numa velocidade de 10 m/s. Em princípio, a peça deveria chegar a sua posição final no tempo 3 segundos. A mensagem enviada para o servidor diz: “O objeto está se movendo para a posição (30,0)”. O servidor sabe a velocidade de movimentação da peça. Suponha-se que a mensagem demore 1 segundo para chegar ao servidor. Quando o servidor receber essa mensagem, ele começa a mover a peça em sua própria simulação na velocidade de 10 m/s, afinal é no servidor que está vivível o estado mais recente do jogo.

Sendo assim, no servidor, a peça chegará à posição (30,0) no tempo de 4 segundos. O servidor então, enviará para todos os clientes, inclusive para aquele que fez o movimento a mensagem recebida. A peça chegará à posição (30,0) em 4 segundos. Todos os clientes, ao receberem a mensagem, calcularão a velocidade da peça necessária para que ela chegue ao seu destino no tempo especificado. Suponha-se então que todos os clientes recebam essa mensagem em 2 segundos. Para os clientes em que o objeto estiver na posição (0,0), a velocidade do objeto será calculada em 15 m/s, o que fará o objeto alcançar a posição (30,0) em 2 segundos. Já para o cliente que gerou o movimento, o objeto já tinha começado a se mover numa velocidade de 10 m/s em 2 segundos e o objeto estava na posição (20,0). Neste

caso, a velocidade é reduzida para 5 m/s, o que fará o objeto também alcançar a posição (30,0) em 2 segundos.

As velocidades nos clientes são calculadas sob a influência da latência, e isso contribui para a redução da dessincronização entre eles. No entanto, um problema com essa técnica é que todos os pontos devem compartilhar o mesmo tempo global. Isso não é algo trivial de se conseguir, mas é possível que os clientes adquiram um tempo estimado do servidor, calculando uma estimativa de RTT⁴(Round Trip Time) entre o servidor e o cliente e fazendo o servidor enviar o seu tempo. Mesmo havendo uma variação entre os tempos, se essa for pequena, a diferença visual do momento de chegada das peças entre os clientes será pequena e possivelmente imperceptível.

c) Eliminando informações desnecessárias: Durante a renderização de ambientes virtuais 3D, um estágio geralmente encontrado é o de *culling*, no qual são eliminadas todas aquelas partes do mundo que não estão visíveis e, portanto, não contribuem para a imagem final. Isso é feito com o objetivo de se reduzir o número de triângulos enviados para a placa gráfica. O mesmo pode ser feito para distribuição. Em geral, não há necessidade de enviar uma informação para um cliente que o jogador não possa perceber.

O que se deseja, geralmente, é enviar para um cliente apenas informações que o jogador pode ver. Por exemplo: informações de movimentação dos jogadores, ambiente e itens visíveis. Diversas estruturas hierárquicas utilizadas para *culling* em computação gráfica podem ser utilizadas também eliminar informações desnecessárias em distribuição (AKENINE-MOLLER, 1999).

Uma maneira de se descobrir rapidamente os objetos próximos de um jogador é usando um grid. Um grid divide o mundo em vários retângulos de tamanho igual. Cada retângulo armazena informações sobre os objetos que se encontram em seu espaço.

Para pegar os objetos próximos a um jogador basta pegar os objetos no retângulo desse jogador e também os objetos nos retângulos considerados próximos. Essa região que define quais objetos estão próximos e possivelmente perceptíveis pelo jogador é chamada de área de interesse. Pode-se encontrar o retângulo em que se acha um jogador por meio das seguintes expressões (grid.x ::= posição.eixoX / larguraRetângulo; grid.y ::= posição.eixoY /

⁴ Round Trip Time

alturaRetângulo;), onde grid.x e grid.y armazenam a posição no grid do jogador, posição.x e posição.y armazenam a posição do jogador e larguraRetângulo e alturaRetângulo armazenam as dimensões dos retângulos do grid.

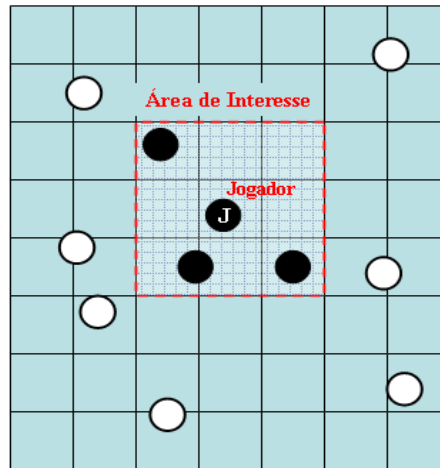


Figura 2-3: Mostra o *culling* usando o grid (SINGHAL e ZYDA, 1999).

A Figura 2-3 mostra o *culling* usando um grid. Todos os personagens próximos ao jogador (os círculos preenchidos) são enviados para o cliente. Já os personagens que estão longe do jogador (círculos não preenchidos) não têm suas informações enviadas para o jogador.

A vantagem do uso do grid está na sua simplicidade e eficiência. Tanto achar os objetos próximos a um jogador, quanto achar a nova posição no grid de um jogador, que se movimentou, são operações de baixo custo computacional. Porém, deve-se tomar cuidado com o uso de objetos grandes, pois eles podem ocupar grande parte de regiões vizinhas e não serem considerados parte dessas regiões. Jogadores movimentando-se nessas regiões podem ver o objeto "surgir do nada", ao mover-se para uma posição no grid próxima à posição do objeto.

Uma outra técnica que pode ser interessante para jogos multi-jogadores, em que a movimentação é geralmente 2D, é a *Quadtree*. Nessa técnica, o mundo é dividido hierarquicamente de 4 em 4 regiões. Primeiro, o mundo é dividido em 4 regiões, depois cada uma dessas regiões é dividida em outras 4 regiões, e esse processo se repete até que algum critério seja atingido.

Nessa técnica, achar os objetos próximos a um jogador tende a ser mais difícil, pois há a necessidade de caminhar hierarquicamente pela *Quadtree* até achar os nós próximos do nó do jogador. No entanto, a *Quadtree* pode ser uma solução adequada quando ocorre uma

grande variação no tamanho dos objetos. Um objeto grande, por exemplo, poderia ocupar um nó superior, enquanto um objeto menor ocuparia um nó abaixo na hierarquia.

A movimentação de objetos também tende a ser mais difícil, devido à complexidade maior da estrutura.

d) Balanceamento: Para conseguir que milhares de jogadores estejam conectados simultaneamente a um mesmo jogo, é necessário que o processamento do jogo seja quebrado entre diversos servidores, e cada servidor seja responsável por processar uma parte desse jogo.

Uma maneira de se fazer isso é quebrar o mundo todo em retângulos, deixando cada servidor responsável por um retângulo e pelo processamento de tudo o que ocorre nesse retângulo. Quando um cliente vai se conectar ao jogo, ele se conecta ao servidor do retângulo em que se encontra e fica se comunicando com esse servidor enquanto estiver nesse retângulo. No momento em que o jogador se mudar para um retângulo de outro servidor, ele se conecta ao novo servidor, e esse se torna o responsável pelo processamento do jogador.

Existe um problema, no entanto, quando esses retângulos que dividem o mundo são estáticos. Muitos jogadores podem se aglomerar em um único retângulo, deixando o servidor responsável sobrecarregado, enquanto outros servidores ficam com pouco processamento. Uma forma de tratar isso é permitindo que as fronteiras dos retângulos possam se mover de maneira a equilibrar o processamento entre os servidores.

Como estabelecer uma conexão demora um determinado tempo (o estabelecimento de uma conexão TCP envolve a troca de 3 mensagens), pode-se tentar estabelecer essa conexão com o novo servidor, antes do jogador de fato entrar em seu retângulo. Para fazer isso, basta expandir as fronteiras dos retângulos.

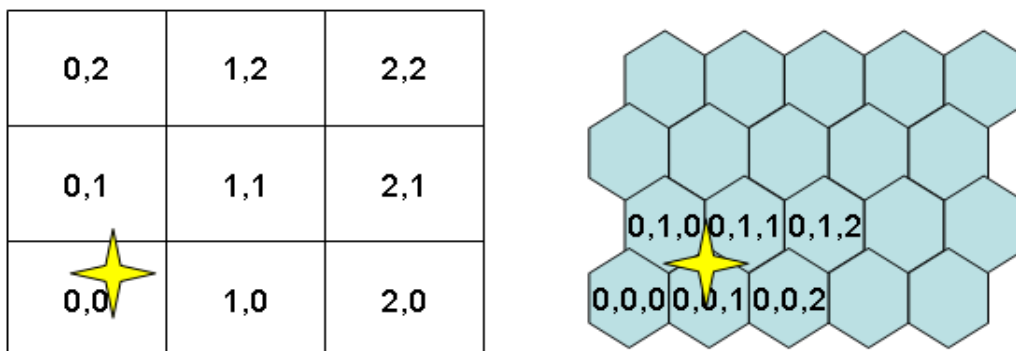


Figura 2-4: Modelo da técnica de particionamento (SINGHAL e ZYDA, 1999).

A implementação de um balanceamento usando essa conexão antecipada deve permitir que os clientes fiquem conectados a mais de um servidor simultaneamente (quando, por exemplo, o cliente estiver em uma fronteira, ficará conectado a pelo menos 2 servidores).

Uma alternativa ao uso de retângulos é o uso de hexágonos para o balanceamento. Nesse caso, o mundo todo é dividido em hexágonos, e cada servidor é responsável pelo processamento de tudo o que ocorre dentro de um desses hexágonos. A vantagem do uso de hexágonos sobre retângulos é que um cliente, ao se aproximar da fronteira, ficará próximo de até 2 servidores, no máximo, enquanto ao usar retângulos, um cliente poderá ficar próximo de até 3 servidores. Ou seja, ao usar retângulos, um cliente ficará conectado com até 4 servidores ao mesmo tempo (quando estiver se movendo por uma fronteira), enquanto, ao usar hexágonos, um cliente só poderá ficar conectado com até 3 servidores. A desvantagem do uso de hexágonos é que existe um custo maior na verificação da ultrapassagem de um cliente por uma das fronteiras.

Alternativa para o balanceamento é fazer o redirecionamento de clientes baseado no tempo que uma mensagem demora para ir e voltar entre o cliente e o servidor, que chamaremos aqui de RTT (*Round Trip Time*). Apenas esse critério não é suficiente, pois um servidor com o RTT bom para um grande número de clientes pode ficar sobrecarregado. Deve ser levado em consideração também o número de clientes e o uso da CPU. Mas a principal limitação dessa forma de balanceamento é que um mesmo servidor pode acabar tratando de clientes próximos a todos os outros clientes e objetos do mundo e isso pode necessitar de mais memória que o servidor pode suportar. Ao usar esse tipo de balanceamento, deve-se aceitar que um servidor pode ter que armazenar informações sobre todos os jogadores do mundo, mesmo que seja responsável apenas por uma fração desses jogadores. Os outros balanceamentos por retângulos e hexágonos permitem a construção de mundos maiores, pois, em geral, um servidor armazenará informação sobre uma fração de jogadores e objetos do mundo.

e) Nível de detalhe: Essa é uma outra técnica que se inspira no nível de detalhe da computação gráfica (SINGHAL e ZYDA, 1999). A idéia é que não há necessidade de se receber muita informação de objetos que não influenciam bastante a experiência do jogador. Em geral, os objetos que mais influenciam a experiência do jogador são aqueles que estão próximos. Logo, a taxa de atualização desses objetos deve ser maior. Objetos longe do jogador podem adquirir uma taxa de atualização menor.

Essa técnica pode ser usada em complemento à técnica de eliminação de informações desnecessárias, por exemplo: Após jogadores distantes e imperceptíveis serem removidos do processamento, pode-se aplicar a técnica de nível de detalhe naqueles jogadores que sobraram, fornecendo menos informações dos jogadores mais distantes e mais informações dos jogadores mais próximos. Por exemplo, aqueles objetos dentro do grid do jogador e vizinhos recebem a maior taxa de atualização, enquanto que os vizinhos dos vizinhos recebem uma taxa de atualização menor.

f) Agregação de mensagens: Outra técnica que permite reduzir o número de bytes enviados em qualquer comunicação é a agregação de mensagens (SINGHAL e ZYDA, 1999). Caso exista uma série de mensagens a serem enviadas para um determinado cliente, é melhor juntá-las numa única mensagem e enviá-la do que enviar uma série de mensagens pequenas. Isso porque mensagens enviadas por UDP adicionam às mensagens 28 bytes de cabeçalho (20 bytes para IP, 8 bytes para UDP) e mensagens TCP adicionam 40 bytes em cabeçalho (20 bytes para IP, 20 bytes para TCP). Logo, quanto menos mensagens são enviadas, menor o desperdício em bytes de cabeçalho para enviar as mensagens.

2.3.9 O uso de ambiente virtuais distribuídos em EAD

Atualmente, a Educação a Distância (EAD) vem assumindo importância cada vez maior no campo da educação formal e na formação e atualização profissional, gerando condições de acesso à educação para pessoas que, por diversos motivos, não puderam obter a formação desejada pelos meios tradicionais de ensino.

A principal característica da Educação a Distância é a separação física entre professor e aluno durante grande parte do curso. Para apoiar este processo utiliza-se algum meio técnico de comunicação, que pode ser desde um simples texto impresso, até conferências por computador com comunicação em duas vias de áudio e vídeo. Neste caso, são utilizados os recursos de telecomunicação assíncrona, vídeo conferência, chat, som e imagens 3D.

A Realidade Virtual (RV) é uma tecnologia nova, que vem ganhando força no mercado, já que oferece ao usuário novas possibilidades de interação com os sistemas computacionais. A navegação em ambientes virtuais favorece a atividade do usuário, que se torna muito mais intuitiva, facilitando a interação homem-computador.

A tecnologia de Realidade Virtual Distribuída vem sendo explorada em diversas áreas do conhecimento, tais como medicina, artes, indústria, cinema e principalmente na educação. A integração da Realidade Virtual Distribuída com a EAD tem gerado novas possibilidades para o aprendizado, criando instrumentos de incentivo à exploração, observação e construção do conhecimento.

Atualmente, as crescentes pesquisas em ambientes virtuais distribuídos têm sido em aplicações educacionais caracterizados com modelos de mundos dinâmicos com complexidade e sofisticação realística por meio dos modelos distribuídos e das diversas entradas/saídas numa escala global.

A utilização das modernas tecnologias de informação e comunicação para o ensino a distância apresenta-se como uma resposta às necessidades de aumento do nível educacional das populações.

A exploração da Internet na EAD destaca-se como uma forte tendência e portanto, será necessário um aumento nos investimentos em pesquisas e desenvolvimento de ferramentas adequadas para o ensino via Internet. Paralelamente, a tecnologia de Realidade Virtual vem se mostrando adequada para apoiar processos educacionais e os resultados das experiências de exploração dos recursos da RV na rede, aliados a constante baixa de preços dos equipamentos, apontam para uma rápida expansão do uso desta tecnologia em vários domínios.

A integração da Realidade Virtual nas práticas de EAD gerará meios de modernizar e dinamizar os recursos e estratégias pedagógicas até então utilizadas, ampliando as possibilidades gráficas e de interação e gerando vivências mais próximas da realidade.

2.4. Considerações Finais

Este capítulo apresentou os conceitos de Realidade Virtual e Realidade Virtual Distribuída. Mostrou que a junção da Realidade Virtual com Sistemas Distribuídos possibilita a existência da Realidade Virtual Distribuída gerando ambientes virtuais distribuídos e relataram as principais características (Capítulo 2.3.1) que um AVD necessita ter.

Todo esse referencial significou a identificação de parâmetros precisos para a elaboração de um ambiente virtual distribuído e apresentou áreas de grandes interesses. Neste contexto optou-se pelo desenvolvimento do estudo de caso “LudosTop”. O “LudosTop” é um jogo educacional 3D constituído de um tabuleiro 4 x 4 (16 casas) e de 16 peças.

O jogo 3D desenvolvido deve fornecer apoio a aprendizagem, e demonstrar a eficiência na comunicação e controle da aplicação multi-jogadores na internet, por meio de técnicas da Realidade Virtual Distribuída. Cabe ressaltar que o foco do trabalho é o uso de técnicas de RV para suporte a jogos multi-jogadores pela Internet.

No próximo capítulo, apresenta-se a um conjunto de jogos, com a finalidade de subsidiar a arquitetura e técnicas aplicadas neste trabalho.

Capítulo 3

3. Trabalhos Relacionados

3.1. Introdução

A seguir, serão relacionados e analisados alguns jogos educacionais dirigidos à utilização de técnicas de Realidade Virtual vinculado à capacidade de colaboração de usuários. Tais análises serão realizadas a partir dos aspectos apontados na fundamentação teórica: da Realidade Virtual e Distribuição. Por meio, das características, possibilidade de interatividade, utilização de técnicas de Realidade Virtual, adequação da arquitetura e, finalmente, uso do compartilhamento vinculado à questão do “acesso simultâneo”.

O objetivo desta análise foi demonstrar relevância e apresentar contribuições com as experiências realizadas, verificando o desenvolvimento de Softwares Educativos.

3.2. Jogos sem Realidade Virtual sem Distribuição

3.2.1 GameKid

O *GameKid* é um jogo educativo específico para o aprendizado de Matemática, a partir da representação gráfica digital 2D. Foi criado na Universidade do Estado do Amazonas (Brasil) (BINDA *et al*, 2005).

O GameKid apóia-se, fundamentalmente em conceitos da Matemática básica, indicado para crianças do Ensino Básico. Quanto aos conteúdos propostos, é um software dirigido a conceitos introdutórios de matemática e destina-se a crianças do Ensino Fundamental. Aborda conceitos tais como a noção de operações básicas (soma, divisão, subtração e multiplicação), medições, raciocínio lógico e temperatura.

Em relação aos aspectos tecnológicos, a plataforma base do jogo é a *engine GameMake*⁵. Para visualização do ambiente do jogo não será necessário ter a *engine* instalada, mas o executável do jogo.

Ao acessar o GameKid, visualiza-se os personagens na tela de apresentação contendo os seguintes botões: iniciar, créditos e sair. Em seguida, a opção de selecionar qual o personagem vai iniciar a partida (Figura 3-1).



Figura 3-1: Tela de seleção de personagem no GameKid (BINDA *et al*, 2005).

Ao clicar a opção iniciar, aparece o ambiente do jogo que contém todos os personagens do jogo que são: Tainá, Ana, Mike, e Johnes. A proposta é de que o usuário explore os cenários de cada personagem, a fim de conhecer os diferentes cenários de cada personagem. O jogo foi desenvolvido com estilo em primeira pessoa para controle do personagem escolhido pelo jogador.

Na fase da personagem Tainá há o jogo da reciclagem, onde é mostrado ao aluno a forma correta de como se tratar o lixo, e a importância da diferença dos materiais utilizados na fabricação dos objetos. Ao separar o lixo de acordo com sua especificação (plástico, papel, vidro) a criança estará ativando o raciocínio lógico matemático. Estará também trabalhando a

⁵ *GameMaker* é uma *engine* para criação de jogos, com recursos de modelagem e programação gráfica.

coordenação motora, pois o usuário terá que colocar os objetos apresentados na tela do computador, na lixeira correspondente ao material utilizado na fabricação de tal objeto.

Na fase do personagem Johnes, é proposto para o usuário problemas matemáticos envolvendo as operações matemáticas de somar e subtrair. A fim de solucionar tais problemas o usuário aciona seus conhecimentos matemáticos de acordo com seu nível de aprendizado.

A personagem Ana tem sua fase desenvolvida num ambiente de temperaturas baixas, portanto o objetivo é a familiarização do aluno com a medição de temperaturas por meio de termômetros que é o instrumento utilizado para fazer tais medições. O aluno terá um termômetro na tela indicando a temperatura ambiente, serão mostradas suposições em que o aluno dirá se está correto ou não, apenas clicando em um dos botões.

Na fase do personagem Mike, que tem como ambiente uma cidade, e devido aos grandes prédios que esta possui abordaremos o tema de medição de altura, largura e área e ainda medidas de distância com transformações métricas (BINDA *et al*, 2005).

Um aspecto negativo do software GameKid é devido a limitação gráfica de efeitos, por meio da sua interface, limitando a imersão. Além de ausência, de recursos para interação entre usuários, tendo necessidade de jogar sozinho.

Tomando como base os parâmetros proposto, pode-se concluir a análise do programa GameKid dizendo que é interativo, atribui características lúdicas ao momento de aprendizagem, propondo um ambiente do imaginário infantil, o que torna adequado para determinada faixa etária. Outro aspecto positivo do programa é a proposta de torná-lo gratuito. Estas características positivas do programa são partes da relação de aspectos a serem buscados no software a ser proposto.

3.2.2 Dominó Matemágico

Dominó Matemágico é um jogo 2D implementado no Macromedia Flash. Foi desenvolvido pelo Grupo de “Tecnologias em Educação e no Ensino da Matemática” da Universidade Federal do Ceará em 2005. (VASCONCELOS *et al*, 2005, *apud* Freitas, 2008)

A finalidade do jogo é trabalhar operações aritméticas (adição) de números naturais, apoiando-se na estratégia de jogos, tendo como público alvo os alunos do Ensino Fundamental.

O objetivo do jogo consiste em formar um “quadrado mágico”, cuja soma dos valores seja cinco. Para alcançar o objetivo, o jogador deverá utilizar as peças de dominó disponíveis no ambiente de jogo (Figura 3.2). Para isso, ele terá de movê-las no sentido horizontal ou vertical, encaixando-as no tabuleiro e formando assim uma matriz de ordem quatro. (VASCONCELOS *et al*, 2005)

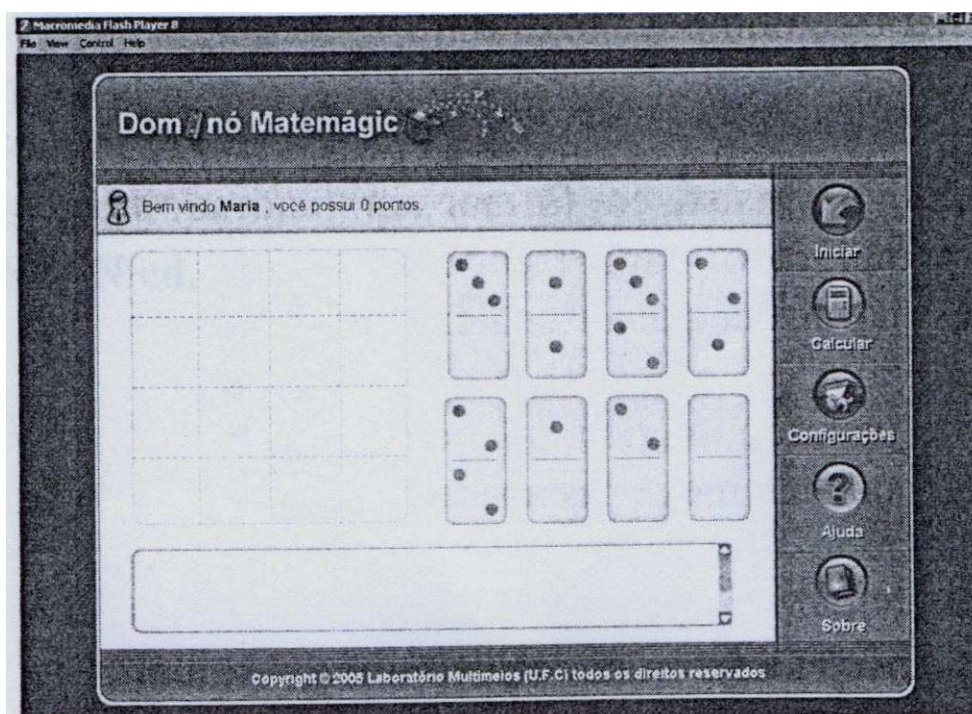


Figura 3-2: Interface gráfica do Dominó Matemático. Fonte (VASCONCELOS *et al*, 2005)

A barra de botões está organizada em: iniciar, calcular, configurações, ajuda e sobre. Essa funcionalidade permite ao jogador iniciar uma jogada, verificar se o quadrado mágico foi montado corretamente, configurar os níveis de dificuldade do jogo e buscar informações sobre o programa, sobre as instruções e sobre os objetivos do jogo.

Outro aspecto interessante do programa é que permite à criança, mesmo sozinha, exercitar o raciocínio matemático.

Finalmente, uma outra característica interessante do programa é a interface, pois todas as opções estão disponíveis na tela principal do jogo, esse aspecto facilitar na jogabilidade e manuseio do jogo.

Tomando-se por base os parâmetros tomados para análise, é possível dizer que o Dominó Matemático envolve aspectos lúdicos; é interativo, informando ao usuário o resultado de suas ações e, finalmente, permite o desenvolvimento do raciocínio lógico matemático com a uma interface interativo contendo as informações de ajuda, configuração e pratica do jogo, ajudando na jogabilidade. Esses aspectos constituem características positivas a serem buscadas no sistema a ser proposto.

3.3. Jogos com Realidade Virtual sem Distribuição

3.3.1 Planeta X

O Planeta X consiste em um jogo educacional criado por Fabio Araújo e André Cordenonsi, por meio da Realidade Virtual não imersiva, no Laboratório de Computação Gráfica da Universidade Luterana do Brasil (ULBRA) em 2003 (ARAÚJO e CORDENONSI 2003).

Desta maneira, tem como objetivo utilizar recursos como visualização em 3D, interação e envolvimento do usuário. Propõe sua utilização como uma ferramenta de auxílio no processo de ensino e aprendizagem multidisciplinar. Pode também ser utilizado por interessados em geral, desde que possuam conhecimentos básicos sobre as leis da física.

O software foi desenvolvido para ser utilizado em navegadores da *Web*. Integrando, as linguagens *JavaScript*, *VRML* e *HTML* apresenta páginas interativas onde estudantes podem navegar e testar seus conhecimentos sobre aspectos do *universo*⁶.

Quanto ao tipo de interface, é composto basicamente de uma tela principal. O *frame* disponibiliza quatro *links* que permitem o acesso às seguintes opções: Estória, Sobre, Jogar e Objetos (Figura 3-3). O *frame* passa a apresentar objetos virtuais, de acordo com a opção selecionada.

No ambiente selecionado, como por exemplo, o do jogo, o usuário pode navegar livremente, visualizando os objetos sob diferentes pontos de vista.

Além dos ambientes de interação com o usuário, o jogo também, possui animações desenvolvidas em VRML, cujo objetivo é informar o usuário o que está ocorrendo durante a sua participação no jogo. Existem quatro animações: uma introdutória, duas que servem para passar ao usuário a passagem de uma fase para outra, e uma animação final, que passa ao usuário a conclusão do jogo. Também existem outras pequenas animações embutidas dentro dos ambientes para deixar o jogo mais perto da realidade.

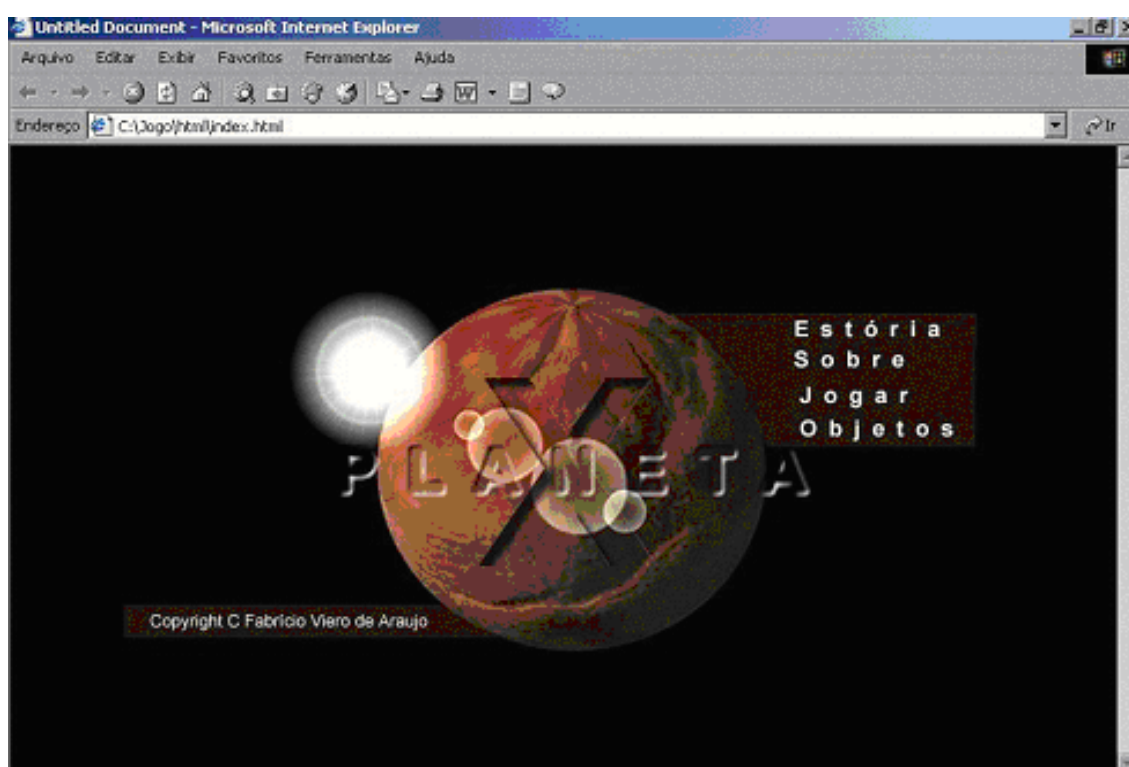


Figura 3-3: Tela do Planeta X (ARAÚJO e CORDENONSI, 2003).

Ao selecionar a opção jogar (Figura 3-3), por exemplo, aparece um novo ambiente dentro da nave. Esse novo ambiente virtual constitui de duas fases. A primeira fase inicia-se com o usuário dentro de uma nave (Figura 3-4; esquerda). A nave encontra-se desligada. O usuário deverá descobrir como ligar a nave para decolar em direção ao Planeta X. Um relógio marca o tempo que o usuário irá gastar para ativar a nave, tal relógio é acionado a contagem no momento em que o usuário inicia as tentativas de ativar o sistema de energia da nave para

⁶ Conjunto de estrelas, planetas, galáxias e outros astros celestes inseridos no sistema espaço-temporal que obedecem às leis da física.

mostrar ao usuário após finalizar esta fase. O sistema de energia se encontra num corredor dentro da própria nave (Figura 3-4; direita).

Ao conseguir ativar o sistema de energia da nave, uma animação é executada. Onde a nave treme e emite um som, indicando que foi ativada. Com o recurso 3D e com som propicia maior grau de imersão no ambiente proposto, porque torna-se mais realista a interação. Após a nave ter sido ativada, o relógio registra o tempo que o usuário gastou para ativa-lá e apresenta ao usuário um novo ambiente (Planeta X), onde se inicia a segunda fase do jogo. O objetivo desta fase consiste em solucionar os enigmas propostos. Para mudar de nível basta responder corretamente tais enigmas, com total de 2 fases.

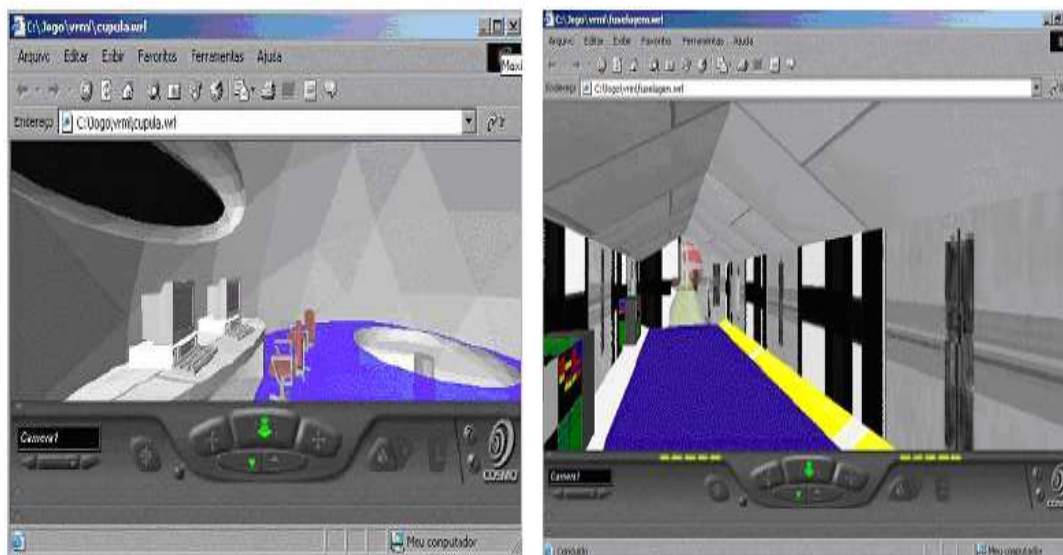


Figura 3-4 : Cenas do Planeta X (ARAÚJO e CORDENONSI, 2003).

Segundo os autores, o jogo possibilita que o usuário desenvolva habilidades de trabalhar com estratégias. A função do usuário (que será incorporado a um agente espacial) é de descobrir, por meio de pistas, o que ocorreu com os colonizadores do “Planeta X”.

O ponto forte do jogo é a interatividade que o usuário tem com os objetos (portas, cadeiras, painéis de controles e etc..) do jogo e os seus ambientes. O jogo está embutido em um conjunto de páginas HTML, que por meio de um navegador auxilia o usuário na integração do jogo.

Uma grande vantagem de utilizar o Planeta X é a possibilidade de se visualizar figuras geométricas tridimensionais em um ambiente 3D, e a facilidade de interação entre jogadores e ambiente, pois possibilita utilizar vários dispositivos de entrada como: data-glove, teclado o

mouse. Outra vantagem é a disponibilidade gratuita via Internet, ou seja, uma acessibilidade maior.

Outro aspecto interessante do Planeta X é permitir uma visualização ativa dos objetos virtuais. Uma maior interatividade possibilitando de interagir com o ambiente proposto (mover objetos, por exemplo), o que implica em respostas em tempo real às ações do usuário.

Tomando-se como base os parâmetros propostos para análise, é possível concluir que o programa Planeta X apresenta dentre os aspectos positivos a sua interatividade com o ambiente (constituído pelo jogo proposto) e utiliza técnicas de Realidade Virtual na construção do software (modelagem dos sólidos geométricos). Isso possibilita boa visualização dos objetos. Esses aspectos positivos serão buscados na construção do software a ser proposto.

3.3.2 Supermercado 3D

O Supermercado⁷ 3D foi desenvolvido em 1998, pela empresa Positivo Informática, tendo como público alvo alunos do Ensino Fundamental. O jogo educacional Supermercado (Figura 3-5) é um ambiente virtual desenvolvido em Visual Basic utilizando técnica de Realidade Virtual em tempo real com a biblioteca Game *Viewer3D* desenvolvida em Visual C++.

Por meio de situações semelhantes às encontradas na vida real, o Software ensina a fazer compras, aplicando conceitos matemáticos, de orientação espacial, direcionados à aprendizagem. As crianças são instigadas a verificar prazos de validade, pesquisar preços e procurar nas prateleiras produtos indicados em uma lista de compras. Ao realizar as tarefas corretamente ganham dinheiro que podem economizar e comprar o que desejarem na seção de brinquedos (POSITIVO, 2008). Além disso, propicia a aprendizagem das 4 operações básicas de matemática, número inteiros, números decimais, frações, proporcionalidade, medidas, orientação espacial, figuras planas e sólidos geométricos. Essas situações de aprendizagens acontecem dentro de um ambiente lúdico que simula um Supermercado, no qual o aluno pode: fazer passeios, fazer compras, comprar com suas economias e ser o atendente do caixa.

⁷ Site do jogo (http://www.positivoinformatica.com.br/site/ed_soft_interna.asp?id_software=59)



Figura 3-5: Tela de Apresentação do Supermercado 3D (POSITIVO, 2008).

O software permite ainda que pais e professores montem um roteiro de compras e personalizem o jogo. O jogo permite que a criança imprima, monte e brinque com o dinheiro e produtos que apareçam no jogo. O Supermercado 3D é um Jogo educativo que possibilita a criança assimilar conteúdos de Matemática, no ato de fazer compras em um Supermercado. É um software aberto, heurístico que apresenta múltiplas opções de atividades em que o aluno vivencia situações reais de uso da Matemática, deparando-se com um leque de "situações-problemas". Tais situações problemas possibilitam a criança colocar em pratica conceitos matemáticos vivenciados na escola.

As compras são feitas por meio de menus, tendo por base as propriedades dos objetos. No caso da representação e construção de roteiro de compra, não é necessário ter um bom domínio sobre o uso das medidas e propriedades dos objetos representados.

Em relação aos aspectos tecnológicos, a plataforma base do jogo é o Visual C++, mas foram inseridos recursos do Direct-X 6.0. Para visualização do ambiente do jogo é necessário um *Plug-In* para Direct-X.

Enquanto design, o jogo Supermercado é simples e pouco atraente em face da grande sofisticação a que chegaram hoje os jogos de computador. Por outro lado, não traz uma opção de interação com outros usuários. De qualquer modo, é um sistema a diversas práticas didáticas.

Tendo por referência os parâmetros proposto para análise, pode-se concluir que o jogo Supermercado é um jogo didático com os controles da GUI efetivamente arquitetado por meio do hardware como: teclado, mouse e vídeo. Esses aspectos constituem características positivas a serem buscadas no sistema a ser proposto.

3.3.3 Escola TRI-Legal

O projeto Escola TRI-Legal consiste na criação de um ambiente virtual, com o objetivo de simular uma escola virtual em três dimensões, onde os estudantes, navegando no ambiente, poderão ter acesso a diversos jogos como instrumentos de educação.

Este AV foi desenvolvido em desenvolvimento no Laboratório de Informática na Educação da Universidade Franciscano (Unifra) em 2005 (VENDRUSCOLO *et al*, 2005).

A Figura 3-6 representa a interface principal por onde o aluno tem acesso aos demais prédios com os conteúdos de aprendizagem a serem estudados.

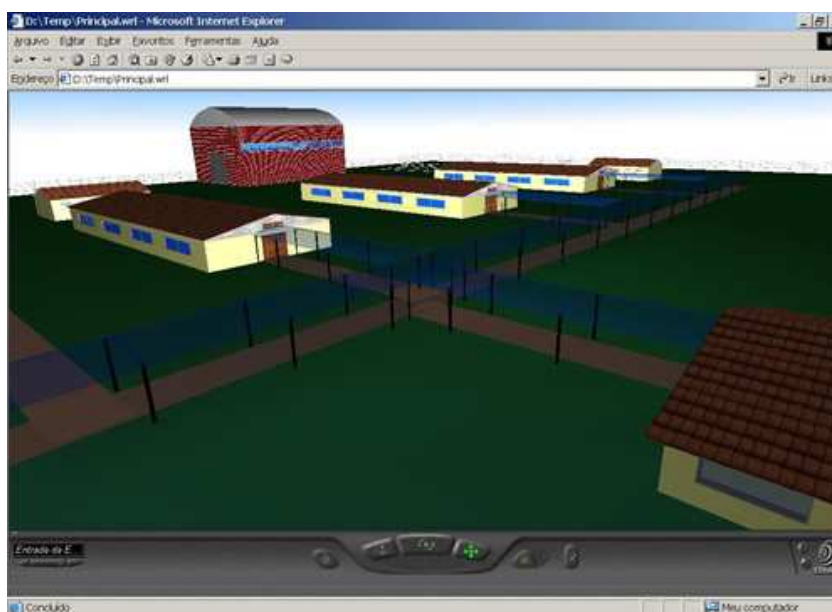


Figura 3-6: Escola Tri-Legal (VENDRUSCOLO *et al*, 2005).

De acordo com os autores, a escola é formada por um conjunto de prédios, cada um com propósitos diferenciados dentro dos conteúdos de aprendizagem. Ao entrar na escola, o aluno estará deslocando-se aos prédios de estudos referentes às disciplinas estudadas, à biblioteca, que tem por objetivo armazenar material de referência sobre todas as matérias ou visitar a lanchonete, refeitório, ginásio de esportes e banheiros que, atualmente, tem finalidade apenas de trazer mais realismo ao ambiente escolar. Por exemplo, no ginásio de esportes (Figura 3-7) o estudante poderá jogar arremessos de basquete e chutes ao gol.

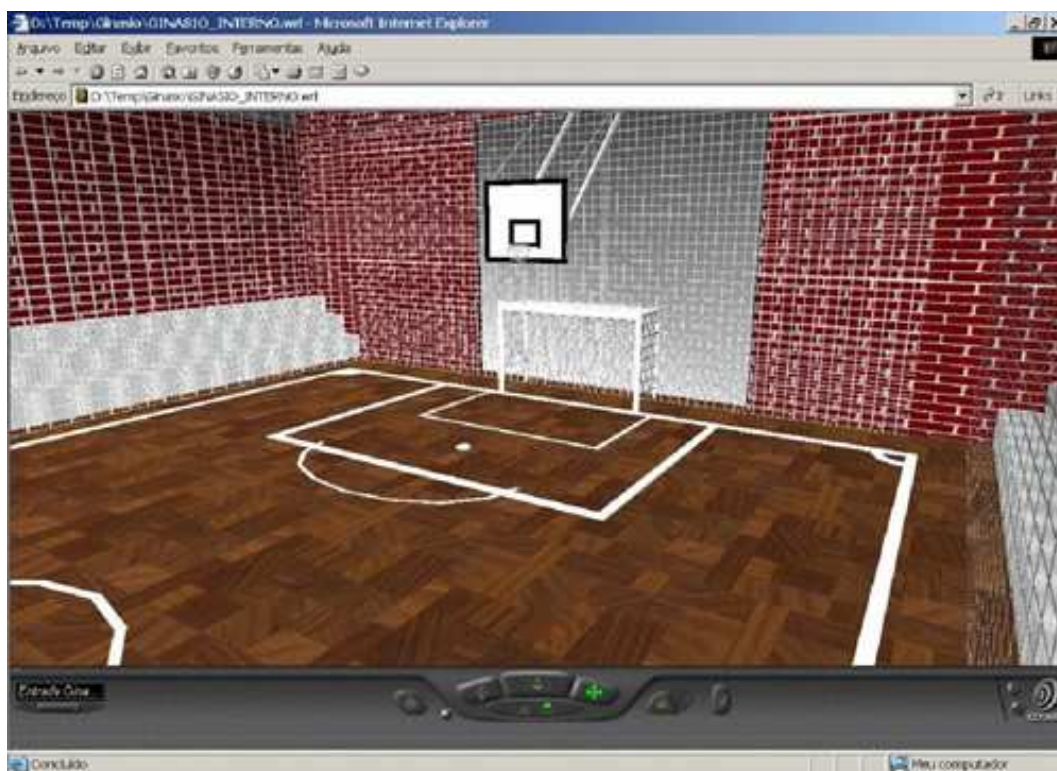


Figura 3-7: Ginásio de esporte tridimensional da Escola Tri-Legal (VENDRUSCOLO *et al*, 2005).

A linguagem utilizada para a implementação da Escola foi a VRML (*Virtual Reality Modeling Language*). A principal característica da VRML é disponibilizar a visualização do Ambiente Virtual na maioria dos navegadores para Web, como Internet Explorer e Netscape (MACHADO, 1995).

Enquanto design, o ambiente educacional da Escola Tri-Legal é bem modelado e traz muitas opções de interação com o usuário. De qualquer modo, o aspecto negativo é que não possui da comunicação entre os participantes, onde seria um excelente ambiente virtual a diversas práticas.

Tendo por referência os parâmetros proposto para análise, pode-se concluir que o ambiente educacional da Escola Tri-Legal é didático, possibilitando que, por meio do ambiente, possa dar continuidade ao seu aprendizado, mesmo sem estar em sala de aula. Mas à interatividade não está bem desenvolvida, onde estes aspectos serão buscados na construção do software a ser proposto.

3.3.4 LudosTop

O jogo LudosTop consiste em um jogo educacional criado por Marlene Freitas, por meio da Realidade Virtual não imersiva, no Laboratório de Computação Gráfica da Universidade Federal de Uberlândia (UFU) em 2008.

O software Ludostop foi feito para ser utilizado em navegadores compatíveis com as linguagens de *JavaScript* e *VRML*. Integrando, as linguagens *JavaScript*, *VRML* e *HTML* apresenta páginas interativas onde estudantes podem jogar.

A sua arquitetura é composta de módulos e sub-módulos: 1) GUI – Interface Gráfica com o Usuário; 2) módulo Internet; 3) módulo Ambientes Virtuais Informativos; 4) módulo Jogo Virtual está dividido em dois sub-módulos internos: Objetos Virtuais e Núcleo Gerenciador de Jogadas.

O primeiro módulo é a interface gráfica com o usuário, disponibiliza os cenários dos Ambientes Virtuais. Estes cenários são compostos por objetos previamente definidos para a visualização, navegação e interação (FREITAS, 2008).

No módulo Internet criou-se uma página HTML que explica o que é o software e também dá acesso a ele. Esta página contém informações importantes para os usuários, sejam estudantes ou professores. O processo de navegação nesta página se dá por meio de botões (Home, Ao Professor, Projeto, Contato, Tutorial e Personagem).

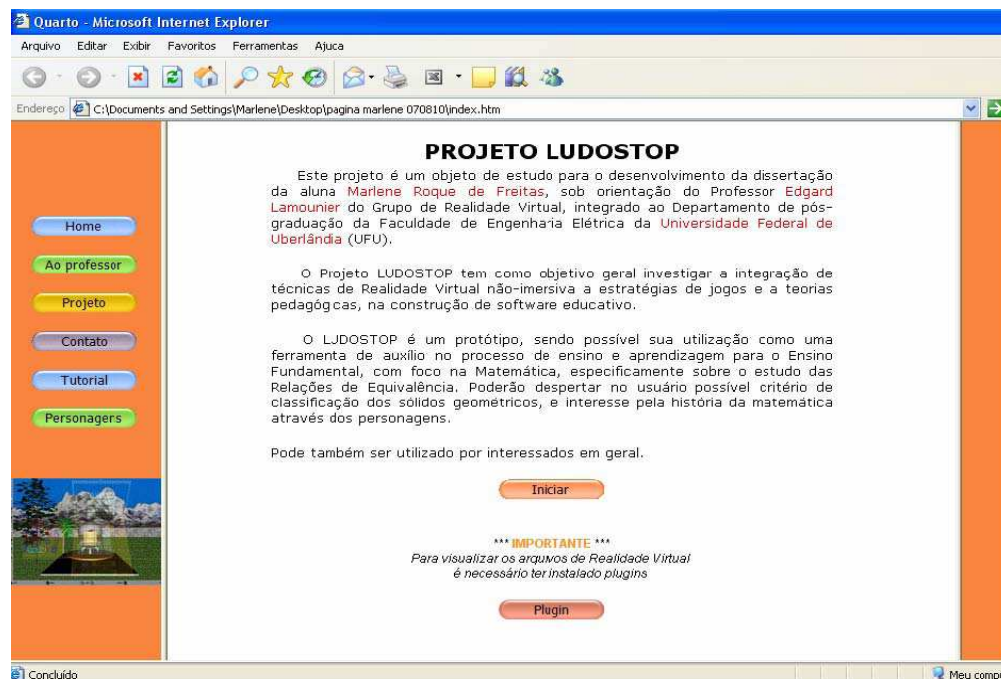


Figura 3-8: Pagina de acesso ao sistema (FREITAS, 2008)

As informações (Figura 3-8) surgem na página à medida que os botões são clicados. Clicando-se o botão “*Home*” aparece um texto explicativo sobre o sistema e mais dois botões: “*Iniciar*” e “*Plug-in*”. O botão *Iniciar* leva a uma nave espacial, ponto de partida da interação do usuário com o sistema. O botão “*Plug-in*” destina-se à instalação do dispositivo para a visualização dos ambientes 3D (relativos ao software LudosTop).

Acionando-se o botão *Ao professor*, aparece um texto cujo objetivo é esclarecer as finalidades pedagógicas do sistema.

Com a inclusão do botão “*Ao Professor*”, este passa a ter a sua disposição todos os objetivos pedagógicos implicados no software, sugestões de como interferir junto ao aluno, bem como considerações sobre a importância dos programas de computador e dos jogos no processo de aprendizagem. Por fim, tem a seu dispor referências bibliográficas: textos sobre o assunto e também outros software educativos considerados interessantes.

O botão “*Projeto*” esclarece sobre as ferramentas utilizadas para a elaboração do sistema, sobre o modo como foi desenvolvido o projeto, bem sobre os seus autores.

O botão “*Contato*” disponibiliza os endereços eletrônicos dos autores, possibilitando trocas de idéias, sugestões e críticas.

O botão “*Tutorial*” mostra como funciona o sistema.

O botão “Personagem” traz a proposta de motivar o usuário a conhecer sobre os personagens relativos à história da matemática, que compõem a lista de codinomes (nick names) a serem escolhidos como identificadores de cada usuário no ambiente do jogo.

Este botão permite o acesso a uma página *Web* tridimensional com informações sobre tais personagens do mundo da matemática, sua vida e conquistas científicas. Isto é apresentado a partir de uma linha do tempo, onde tais personagens podem ser acessados de modo cronológico. Na Figura 3-9 demonstra cada personagem relatado num livro. Este livro pode ser manuseado pelo usuário. As páginas contêm imagens e informações resumidas sobre as descobertas destes cientistas.

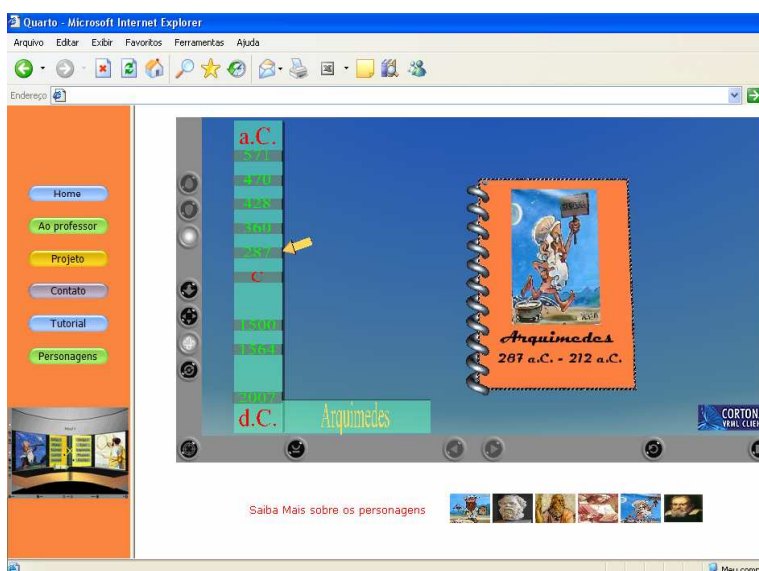


Figura 3-9: Botão personagens - linha do tempo (FREITAS, 2008).

O módulo (Figura 3-10) Ambiente Virtual Informativo corresponde a uma nave espacial que foi modelada a partir da linguagem VRML 2.0. Os objetos foram construídos pelos seguintes processos, de acordo com o grau de complexidade: a partir de primitivas geométricas (cilindros, cones, paralelepípedos e esferas), cujas linhas de código de representação já estavam disponíveis; realização de operações booleanas (adição, subtração e interseção) com estas primitivas; transformações de figuras planas em objetos tridimensionais pela técnica de extrusão⁸ (FREITAS, 2008).

⁸ A técnica de extrusão é empregada para a modelagem de ambientes ou modelos complexos aplicando cortes transversais corrigindo defeitos de modelagem.



Figura A

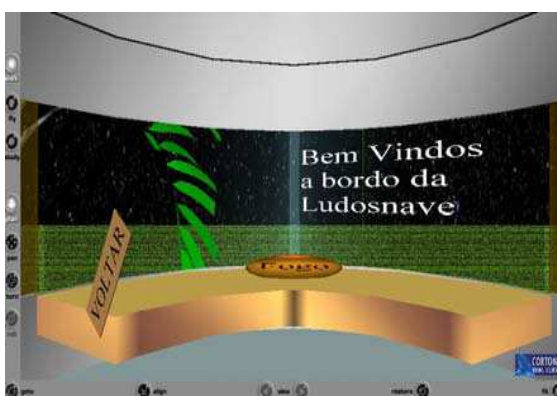


Figura B

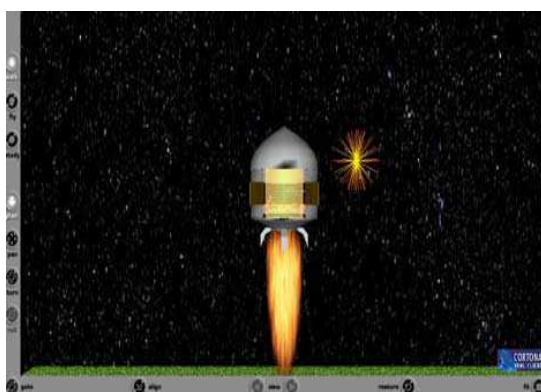


Figura C

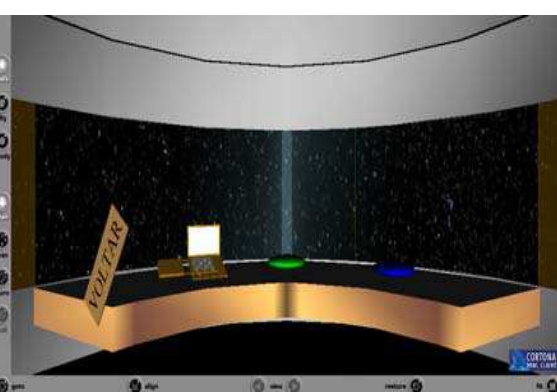


Figura D

Figura 3-10: Procedimentos iniciais utilização do sistema (FREITAS, 2008).

O botão “Ajuda” permite que o usuário tire algumas dúvidas com relação ao uso do sistema. O botão “Personagens” permite visualizar imagens de personagens da história da matemática, possibilitando que se conheça sua vida e obra. As imagens são mostradas na parte dianteira da nave e demais informações por meio de páginas HTML. Tais personagens farão parte das opções para escolha do identificador do usuário no ambiente do jogo (apelido).

Com a Figura 3-11, mostra a escolha do apelido são apresentadas duas placas com quatro opções de apelido cada uma. Ao selecionar uma opção é apresentada a imagem correspondente ao personagem escolhido.



Figura 3-11: Opções de apelido e setas para navegar nas opções (FREITAS, 2008).

O botão “Jogo” permite que o usuário escolha o nível do jogo, o apelido (personagens pré-selecionados) e as cores das peças. A escolha do nível é obrigatória. A escolha do apelido e das cores é opcional. Tais informações são enviadas para módulo Jogo Virtual.

Definiu-se três partidas para cada jogada. Durante as partidas, os jogadores tentam alcançar o objetivo, que é alinhar peças com um ou mais atributos comuns. Alcançado o objetivo a partida é finalizada. O vencedor do jogo é quem ganha o maior número de partidas.

Pode acontecer, todavia, que se encerrem as peças sem que nenhum jogador tenha vencido. Neste caso, a partida é também considerada encerrada e o sistema acusa empate técnico (na partida). Neste caso não tem pontuação.

Os resultados de cada partida (mesmo os empates) e o resultado final do jogo são disponibilizados numa tabela (placar).

Ao jogar, é possível escolher entre dois níveis.

No nível I, o jogador escolhe uma peça e a posiciona em uma casa do tabuleiro; passa a vez para o adversário, que posiciona outra peça em outra casa. O objetivo, como já foi afirmado, é alinhar peças com pelos menos um atributo, assim, segue até o final da partida.

No nível II, que é um pouco mais complexo, o jogador 1 escolhe uma peça para o jogador 2. O jogador 2, tendo esta peça em mãos, a posiciona no tabuleiro; em seguida, escolhe outra peça para o jogador 1. O jogo segue dessa forma até o final da partida.

No que se refere à questão de quem começa a jogar, isso é definido por meio do acordo entre os jogadores. Decidido quem irá iniciar a partida, define-se que esta pessoa será o Jogador I.

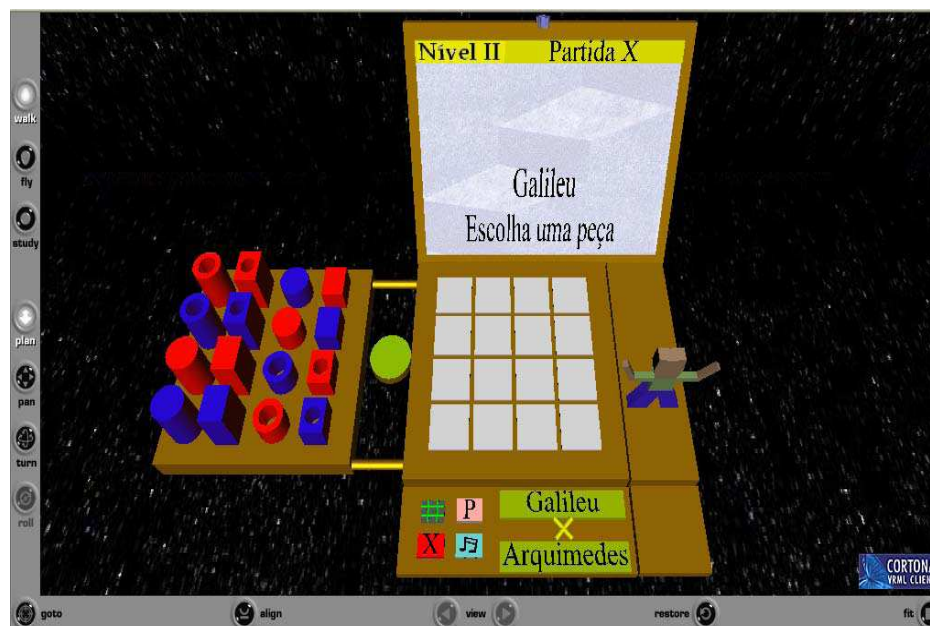


Figura 3-12: Laptop – ambiente para jogar (FREITAS, 2008).

Na Figura 3.12, é possível observar um exemplo de iniciar a partida, onde o jogador “Galileu” começará a selecionar a peça ao seu adversário.

O software “LudosTop” possui aspectos pedagógicos já fundamentos em relação à Educação de Matemática, além de considerar a tecnologia utilizada (linguagens Javascript e VRML) visou criar um software bastante acessível tanto do ponto de vista operacional quanto aquisitivo.

O ponto forte do jogo é utilizar técnicas de RV que, integradas a uma fundamentação pedagógica consistente, permitissem a elaboração de uma arquitetura lúdica, figurativa, atraente, interativa e adequada para a faixa etária envolvida.

Finalmente, uma outra característica interessante é a arquitetura do jogo dividido em módulo, torna-se a manutenção da aplicação mais efetiva.

Tomando-se por base os parâmetros tomados para análise, é possível concluir que o software LudosTop apresenta aspectos extremamente positivos como mencionados acima, quanto a sua aplicabilidade, utilização de técnicas de RV na construção. Além da arquitetura modular trazem vantagem desde a implementação como na operação. Devido a esses pontos o LudosTop foi a motivação para o desenvolvimento do atual trabalho.

3.4. Jogos com Realidade Virtual com Distribuição

3.4.1 EducaTrans

O EducaTrans é um jogo educativo para a educação no trânsito, foi criado na Escola Politécnica da Universidade de São Paulo (ASSIS *et al*, 2006)

O jogo consiste em uma simulação do fluxo de pedestres, motoristas; e ciclistas que partem de um mesmo ponto e buscam atingir o mesmo destino no menor tempo e com o menor risco à sobrevivência, tendo como público-alvo alunos do Ensino Fundamental e Ensino Médio.

O jogador deve escolher qual papel estará desempenhando no decorrer do jogo, por exemplo, o de pedestre, de ciclista ou de motorista. Sabe-se que cada personagem atua de modo diferente no trânsito, por isso, o jogador deverá estar atento às leis de trânsito as quais deverá respeitar durante o jogo.

Desta forma, o jogo EducaTrans tem objetivos cognitivos nos níveis de conhecimento (leis de trânsito, sinalização de vias), compreensão (interpretação das pistas para descoberta do tesouro) e avaliação (o aprendiz é capaz de confrontar o seu desempenho com os demais participantes por meio das decisões tomadas e recompensas e penalidades atribuídas) e também objetivos afetivos (estimula a imaginação e autoconfiança do aprendiz). No jogo EducaTrans a estratégia mais adequada é a orientada por descoberta, onde o papel do professor é de orientador durante o jogo.

A justificativa do jogo são os acidentes de trânsito causados por imprudência de pedestres e ciclistas, por falta de conhecimento, de orientação e de familiarização com leis e placas de trânsito. Por outro lado, há uma carência de jogos educacionais que trabalhem a Educação no Trânsito. A disponibilização do jogo EducaTrans visa suprir esta demanda e contribuir com as ações, palestras, aulas, ao ser adotado nas escolas, nos telecentros e nas residências.

O cenário do jogo (Figura 3-13) apresenta um quarteirão com vias de trânsito (ruas, avenidas, rodovias), calçadas, faixas de pedestre, rotatória, semáforos para pedestres e veículos, placas de trânsito de indicação, de advertência e de regulamentação.

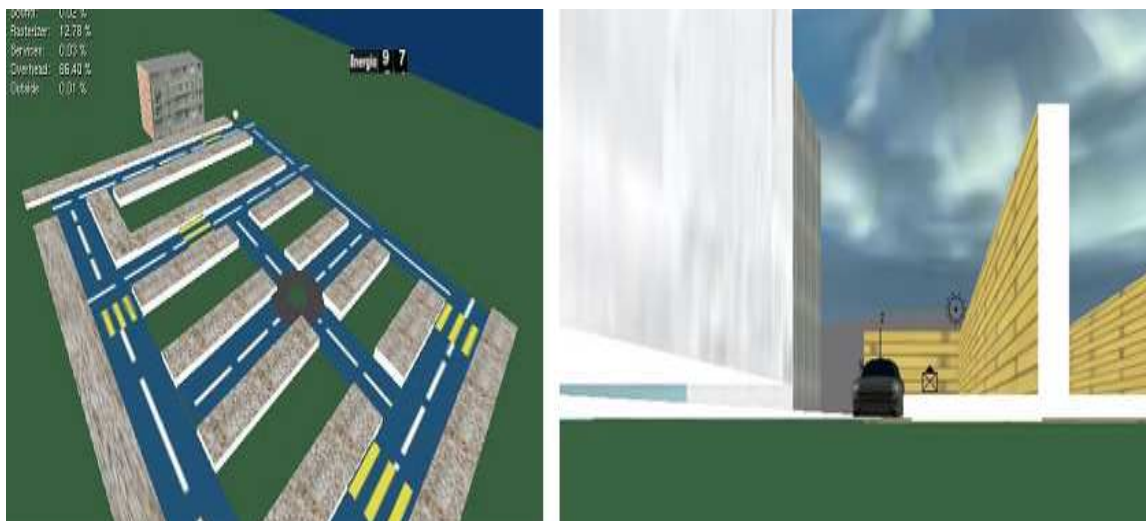


Figura 3-13: Cenário do EducaTrans (ASSIS et al, 2006).

Em relação aos aspectos tecnológicos, o jogo EducaTrans procura envolver o jogador em um ambiente imersivo com técnicas de Realidade Virtual, apresentando um ambiente dinâmico.

A construção do software foi por meio da engine *fly3D*, tendo por base as bibliotecas *OpenSceneGraph*; com a utilização do Blender3D para modelagem; e utiliza algoritmos genéticos como mecanismo de evolução dos pedestres autônomos e oferece um ambiente tridimensional que reproduz um cenário real com regras de trânsito bem definidas.

O sistema apresenta um *design* criativo à oferece um ambiente multi-usuário, com nível alto de interatividade aos usuários devido aos cenários e aos personagens permitem se deslocar no cenário usando o teclado. Setas de direção permitem avançar, recuar, virar a direita e virar a esquerda.

Outro aspecto é a interação do usuário e competidores: o jogador controla apenas um personagem da “corrida”. Os demais personagens são controlados no modo multi-jogador por outros usuários ligados em rede, e no modo jogador-simples, os demais personagens que participam da “corrida” são gerados e controlados por mecanismos de Inteligência Artificial. O modelo de comunicação cliente-servidor, desempenha o papel de um sistema multi-jogador, simulados por utilizado Algoritmo Genético (Algoritmo de Inteligência Artificial).

Tomando-se como base os parâmetros propostos para análise, é possível concluir que o jogo EducaTrans apresenta, enquanto aspectos positivos o fato de utilizar técnicas de Realidade Virtual na construção do software, é uma arquitetura cliente-servidor que por meio

da Internet usuários possam jogar. Esses aspectos positivos serão buscados na construção do software a ser proposto.

3.4.2 Peloton

O Peloton um ambiente virtual distribuído de simulação ciclística, onde vários usuários conectados em diferentes computadores sobre a Internet são capazes de pedalar simultaneamente em tempo real usando diferentes dispositivos de entrada e saída, conforme a Figura 3-14; esquerdo.

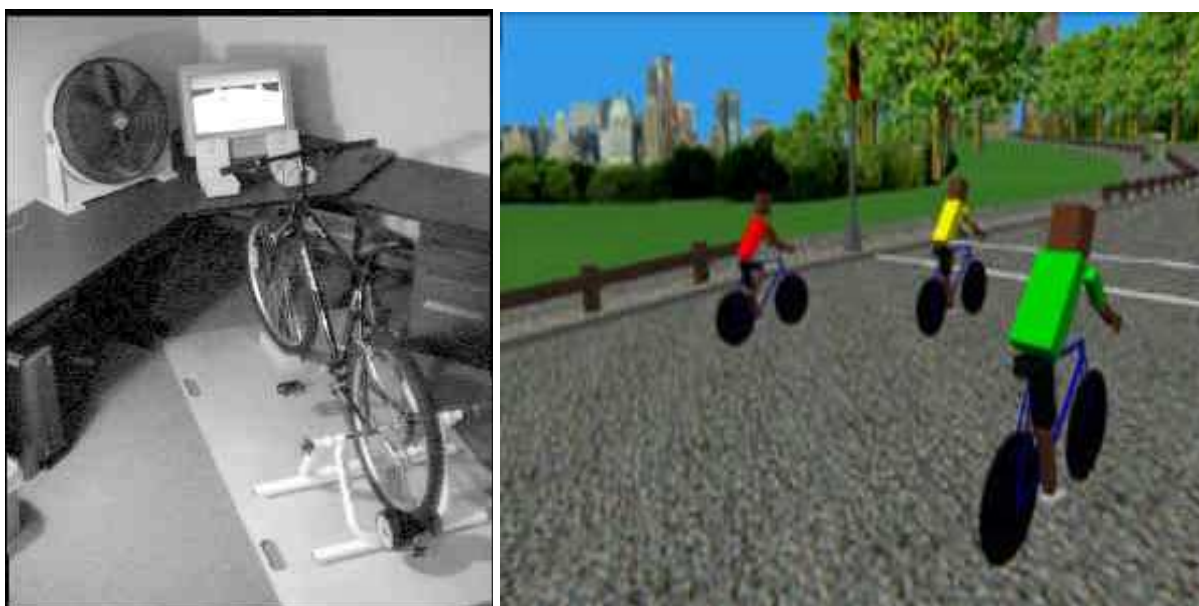


Figura 3-14: Dispositivo de entrada e saída do Peloton (CARRARO et al, 1998).

O sistema foi desenvolvido pelo *Bell Laboratories*, Instituto de pesquisas da Lucent. (CARRARO *et al*, 1998). O AVD Peloton consiste de uma simulação multiusuária esportiva, modelado em *VRML*, com uma paisagem urbana para treinamento de atletas.

A solução do Peloton possui uma arquitetura cliente-servidor complexa, devido o grande número de integração entre sistemas, protocolos e linguagens. Na aplicação cliente é dividida em duas partes: *hardware* e *software*. O hardware da simulação é composto de três dispositivos: bicicleta, monitor e ventilador. A parte lógica (*software*) contempla protocolos de comunicação e linguagens. Os softwares controlam desde a comunicação, falhas, interação e sessões na simulação conforme definido na arquitetura (CARRARO *et al*, 1998).

A arquitetura dispõe na parte cliente de um núcleo de gerenciamento formado de três componentes: *Simulation Manager*, *Device Manager* e *local Session Manager*, o gerenciador controla equipamentos (bicicleta, imagens de vídeo e ventilador) por meio de sensores enviando informações via socket ao gerenciador. Uma das tarefas mais importantes do gerenciador é o controle das requisições *on-line* ao servidor com o protocolo *IIOP*⁹, onde enviar e receber pacotes de atualizações que são encaminhadas ao *EAI*¹⁰ integrado com *LiveConnect* atualiza o ambiente virtual em *VRML* no navegador, mantendo o controle de cada sessão.

Enquanto a aplicação servidora consiste do armazenamento das modificações das alterações e gerenciamento de sessões apoiado em um repositório de cenas acessado via *Web Server*.

O Peloton apresenta resultados aceitáveis da solução, com aspectos interessantes referentes à integração com protocolos de comunicação. Embora possua socket, IIOP e Web Server as mensagens são transmitidas com boa velocidade de ~90 m/s para transmissão dos seus pacotes ao servidor e mais ~75 m/s para replicação aos clientes (CARRARO *et al*, 1998).

Tendo por referência os parâmetros propostos para análise, pode-se concluir que o Peloton possui excelentes aspectos tecnológicos, um exemplo explícito é uma arquitetura modular com vários nós de integração, embora adaptado à heterogeneidade dos programas implementados em sua especificação atuando distribuído sobre a internet. Tais propriedades são positivas para tornar-se referência para a construção do sistema proposto. Entretanto não existe controle sobre a sincronização entre clientes com diferentes larguras de banda, assim aqueles clientes com maior latência com o servidor podem ficar bastante dessincronizados e enfrenta o problema de gerenciamento do estado compartilhado, além de usar protocolos lentos como IIOP. Tal propriedade é negativa para tornar-se referência na construção do sistema proposto. Embora o Peloton possua um satisfatório funcionamento durante a renderização nas simulações não foi implementada nenhuma técnica para melhorar o tempo para uma

⁹ Internet Inter-ORB Protocol é um protocolo que faz possível para programas distribuídos escrito em diferentes linguagens de programa comunicar sobre a internet.

¹⁰ External Authoring Interface

mensagem chegar ao destino, entretanto poderia ser melhorado os tempos de transmissão, por meio de técnicas de Ambientes Virtuais Distribuídos.

3.4.3 Laboratório Multidisciplinar Virtual em RV

O Laboratório Multidisciplinar Virtual (LMV) é um sistema educativo 3D distribuído com RV criado com o uso da biblioteca OpenGL usando computação distribuído heterogêneo com CORBA (*Common Object Request Broker Architecture*). Foi desenvolvido no Laboratório de Computação Gráfica da Universidade Federal de Uberlândia em 2006 (RIBEIRO, 2006).

Por meio de técnica de RV não imersiva, o software é formado por dois ambientes virtuais distribuídos. Essa formação é composta de um Ambiente Virtual Biologia (AVB) e um Ambiente Virtual de Química (AVQ).

O AVB (Figura 3-15) foi modelado com uso da biblioteca gráfica OpenGL, sendo necessário o uso da linguagem de programação Delphi 6.0 TM para compilação do código e implementação da interação e das animações, onde encontra-se o objeto sol, modelado em OpenGL como uma esfera. Este objeto possui um código de seleção permitindo ao usuário selecioná-lo em qualquer localização ou posição que o mesmo esteja no mundo. A seleção do objeto que é feita por meio do clique do mouse possibilita a alteração na temperatura e luminosidade do mundo de Biologia. A temperatura do mundo de Biologia assume valores variando de 15°C à 35°C, que, ao serem alterados pelo usuário, inicia uma animação nas folhas das árvores, aumentando ou diminuindo (RIBEIRO, 2006, p.107).



Figura 3-15: Tela do Ambiente virtual de Biologia (RIBEIRO, 2006).

Para formar o LVM, também foi criado o AVQ (Figura 3-16), para simular a animação da Fotofosforilação acíclica foi criada, primeiramente, a estrutura vegetal especializada no processo da fotossíntese, conhecida por cloroplasto, representada por estroma, membrana interna e externa que foi modelada por um software de modelagem 3D (3D Max) e convertidas por meio de um loader criando um arquivo texto com posições geométricas e outras informações (índice dos triângulos, coordenadas x, y e z para os vértices, normais e texturas, informações sobre materiais, luzes e câmeras) do objeto.

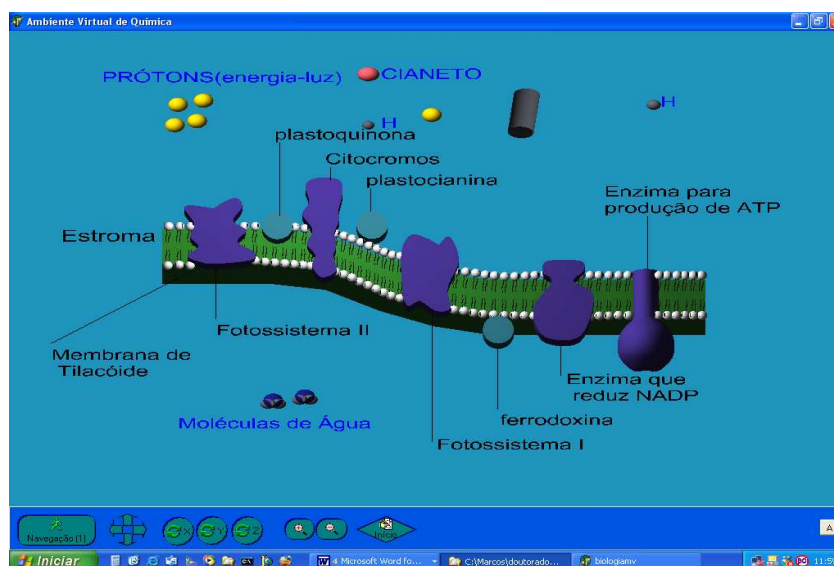


Figura 3-16: Tela do Ambiente Virtual de Química (RIBEIRO, 2006).

Na implementação da distribuição e comunicação entre os AVs foi usada a linguagem de programação Borland Delphi 6.0 Interprise TM com suporte ao Inprise Visibroker 4.0, usando o Inter-ORB Protocol (IIOP), que se encontra uma camada acima do TCP/IP.

A arquitetura resumidamente, verifica-se que:

- Cada ambiente virtual pode ser executado separadamente e possui interação dentro do próprio mundo. A interação é sempre disparada por um determinado evento. Se o par de ambientes estiver ativo em uma rede, além das respostas no próprio ambiente (animação) haverá também em outro ambiente o disparo de eventos. Estes parâmetros são repassados para um Cliente/Servidor que é responsável por localizar um outro Cliente/Servidor na rede ou na própria máquina;

- O usuário interage com o ambiente virtual 1, de tal forma que as informações são capturadas pelo objeto 1. O objeto 1 (cliente/servidor) localiza na rede o objeto 2 (cliente/servidor) com a ajuda VisiBroker (padrão CORBA) que responderá a requisição do objeto 1, que repassa as informações ao ambiente virtual 2 e apresenta a nova interface ao usuário e vice-versa.

O ponto forte deste projeto foi analisar e implementa os requisitos dos AVDs, e assim medir à eficácia destas técnicas, que vão das características de comunicação em rede, modelo de visão, modelo de dados, gerenciamento da computação até o comportamentos dos objetos (RIBEIRO, 2006).

Esse trabalho é inovador em relação ao processo multidisciplinar e tecnologico, pois demonstra como resultado final o protótipo que usando o CORBA como plataforma de distribuição, modelo de visão assíncrona, gerenciamento da computação com distribuição parcial e modelos de dados cliente-servidor como melhor arquitetura de distribuição.

Tendo por referência os parâmetros propostos para análise, pode se concluir que o LVM possui interessante todos os aspectos tecnologicos como arquitetura e técnicas são positivas para torna-se referência para a construção do sistema proposto.

3.4.4 Laboratório Multidisciplinar Virtual em RA

O Laboratório Multidisciplinar Virtual em Realidade Aumentada (LMVRA) é uma continuação do sistema educativo 3D distribuído com RV, criado sobre a biblioteca OpenGL usando computação distribuído heterogêneo com CORBA (*Common Object Request Broker Architecture*). A LMVRA foi desenvolvido no Laboratório de Computação Gráfica da Universidade Federal de Uberlândia em 2008 (SILVA, 2008).

Nesta arquitetura, CORBA é usada para suportar a comunicação entre vários (n) computadores. Cada computador pode ter um tipo de ambiente, Biologia ou Química. Haverá apenas um ambiente de Química e “n” ambientes de Biologia (Figura 3-17).

Sempre haverá um objeto servidor que se encarregará de gerenciar a distribuição dos ambientes. Este servidor poderá ser uma aplicação dedicada, ou até mesmo o próprio objeto cliente poderá ter papel de servidor em algum momento.

As comunicações são efetuadas por meio da camada CORBA (comunicação cliente/servidor e comunicação cliente/cliente por meio do canal de eventos).

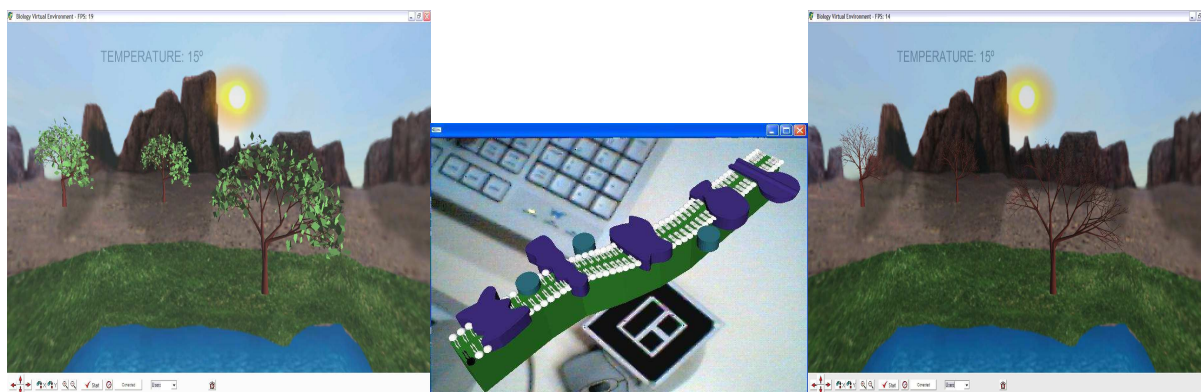


Figura A: Inicia Biologia

Figura B: Processo Química

Figura C: Finaliza Biologia

Figura 3-17: Exemplo de funcionamento do Sistema (SILVA, 2008).

Neste trabalho a aplicação servidora será sempre o ambiente virtual de Química, que usa RA na visualização. Toda aplicação que for iniciada na rede sempre procurará por um servidor, desta forma, haverá sempre apenas uma aplicação do ambiente de Química. Todos os outros ambientes (Biologia) se conectarão ao servidor (Química) como clientes.

Qualquer interação no ambiente de Química será replicada para todos os clientes. Por outro lado, as interações realizadas por usuários Biologia alterarão a realidade do ambiente de

Química. O sincronismo entre os ambientes de Biologia será proporcionado por uma comunicação cliente-cliente mediante o uso do Canal de Eventos (Serviços CORBA) (SILVA, 2008).

Como demonstrado (Figura 3-17) no início da aplicação (primeiro cliente), o ambiente começa a interagir com o usuário a partir do estado zero (Biologia: temperatura 15 graus, árvores com folhas pequenas e luminosidade baixa; Química: Cianeto fora da cadeia, processo da fotofosforilação acíclica paralisado). A responsabilidade do armazenamento destes estados é do servidor (ambiente de Química). Sempre que houver alguma interação, o valor do estado do ambiente será modificado e enviado ao servidor, que replica esta informação para todas as cópias.

Enquanto o design, o LVMRA é um sofisticado ambiente virtual usando as heranças do seu antecessor LVM. Pois o aspecto interessante e a heterogeneidade que o CORBA fornece, mais o software necessita ser melhorado em aspecto de balanceamento de carga, senão a aplicação servidor será um limitador da aplicação. Além que o software somente está preparado para funcionar em LANs, pois com a utilização da Internet o software seria completamente interativo.

3.5. Tabelas de Comparação dos Jogos Avaliados

A análise dos sistemas aqui relacionados ao trabalho, e que fundamentam as decisões que foram tomadas em relação ao projeto.

Recursos	Sem suporte a RV sem distribuição		Com suporte a RV sem distribuição			Com suporte a RV com distribuição			
	<i>GameKid</i>	<i>Dominó Matemágico</i>	<i>Planeta X</i>	<i>Supermercado</i>	<i>LudosTop</i>	<i>EducaTrans</i>	<i>Peloton</i>	<i>LVM</i>	<i>LVMRA</i>
AVDs	<i>GameKid</i>	<i>Dominó Matemágico</i>	<i>Planeta X</i>	<i>Supermercado</i>	<i>LudosTop</i>	<i>EducaTrans</i>	<i>Peloton</i>	<i>LVM</i>	<i>LVMRA</i>
Ambiente	2D	2D	3D	3D	3D	3D distribuído	3D distribuído	3D distribuído	3D distribuído
Simulação/Tema	Matemática	Matemática	Jogos	Educação	Jogo	Transito	Esporte	Biologia/Química	Biologia/Química
Recursos para Interação	-	-	Efeitos	Não	Restrito	Mouse	Dispositivos & Sensores	Mouse	Marcadores, Mouse e Video
Modelo de Visão	Não	Não	Não	Não	Não	Assíncrona	Síncrona	Assíncrona	Assíncrona
Comunicação	Local	Local	Local	Local	Local	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor
Categoria	Monousuário	Monousuário	Monousuário	Monousuário	Monousuário	Multiusuário	Multiusuário	Multiusuário	Multiusuário
Gerenciamento d' Computação	Não	Não	Não	Não	Não	Replicação Total	Replicação Parcial	Distribuição Parcial	Replicação Parcial
Técnicas RVD	Não	Não	Modelagem	Modelagem	Modelagem	Parcial	Parcial	Sim	Sim
Transmissão de Mensagens	-	-	-	-	-	Socket via Fly3D engine	Socket, IIOP, Web Server	CORBA	CORBA
Modelagem	EngineMaker	Flash	VRML	Direct X	VRML	Engine Fly3D	VRML	3D Max	3D Max
Modelo de Dados	Não	Não	Não	Não	Não	Compartilhada	Compartilhada	Compartilhada	Compartilhada
Projeto Online	Não	Não	Não	Não	Sim	Sim	Sim	Não	Não
Largura de banda	Não	Não	Baixa	Não	Baixa	Alta	Alta	Alta	Alta

Tabela 3-1: Tabela de comparação entre ambientes virtuais educativos.

3.6. Considerações Finais

Conforme descrito neste capítulo 3, os trabalhos relacionados demonstram estudos, características e técnicas aplicadas em ambientes de aprendizagem virtual.

Seguindo a caracterização da especificação de um ambiente virtual distribuído, por meio da comparação dos softwares mencionados anteriormente foi satisfeita uma variedade de elementos. Primeiramente com à questão da arquitetura, foi pesquisado comunicação em rede, visão, modelo de dados, gerenciamento da computação e comportamento dos objetos.

Para avaliar a comunicação em rede foi baseada em quatro ambientes: EducaTrans, Peloton, LVM e LVMRA. Essas aplicações possuem estrutura cliente-servidor com diversas características as principais são: o suporte para interação em tempo real pela Internet onde o EducaTrans tem melhor condição de resposta por usar técnica de RV desde a modelagem e uso de algoritmos genéticos (IA) com esquema de *broadcast* no processo de comunicação entre cliente-servidor. Ao contrario dos outros sistemas avaliados, não foram implementados com esses recursos.

O paradigma de visões do usuário de um AVD, pode ser observado claramente que a perspectiva assíncrona apresenta maior controle individual, mostrado pelos EducaTrans, LVM e LVMRA testado em redes de longa distância tal como EducaTrans.

Com o modelo de dados usado pelos ambientes virtuais distribuídos EducaTrans, Peloton, LVM e LVMRA foi o compartilhado distribuído devido os requisitos do sistema e demonstrou aplicável independente da tecnologia.

Em relação ao uso do gerenciamento da computação, o LVM apresenta um caso de sucesso com a sua arquitetura usando uma distribuição parcial, pois os estados dos objetos são atualizados localmente e não tendo a duplicação de esforço computacional como resultado do trabalho de Ribeiro, (2006).

Outro ponto fundamental e o comportamento dos objetos foram observados as entradas (via dispositivos de entrada) não-determinísticas para prever a carga do sistema latência e largura de banda. Nas aplicações comparadas não foram planejados a característica mencionada, portanto no software proposto será um aspecto a ser executado.

Pois por meio dos levantamentos de informações foi possível medir o atual nível das aplicações de aprendizagem que faz uso de tecnologias de distribuição ou não.

Além de técnicas de RVD juntamente com integração à da modelagem usando o 3D Max e a plataforma OpenGL para implementação e parametrização. Pode-se verificar que os sistemas multi-usuários mostraram ausência de recurso de interação, ou seja, chat, e-mail, voz ou outra maneira de comunicação.

Pode-se concluir que as aplicações estão mais sofisticadas em nível de arquitetura, porém é necessário tornar-las mais eficientes em comunicação, melhorando completamente os ambientes virtuais, por meio de transmissões mais rápidas e minimizando atrasos na comunicação. Por meio das comparações efetuadas podemos observar que os elementos citados serão adotados com a finalidade de subsidiar o desenvolvimento do sistema proposto.

No próximo capítulo, será especificada a arquitetura do sistema proposto.

Capítulo 4

4. Arquitetura e Funcionamento do Sistema

4.1. Introdução

O presente capítulo descreve a arquitetura do sistema proposto e seu respectivo funcionamento. Além disso, são apresentadas as ferramentas que serviram de base para a sua elaboração (*Torque Game Engine* e a *OpenGL API*).

Este capítulo aponta também uma breve descrição do jogo Quarto. Tal jogo compõe o estudo de caso escolhido para demonstrar a aplicação de técnicas computacionais que associadas à Realidade Virtual Distribuída, possibilitem o desenvolvimento do software proposto na Internet.

4.2. Tecnologia de Apoio

Nesta seção, são apresentadas as tecnologias de apoio que foram selecionadas para a implementação do sistema proposto. Posteriormente, será mostrado como essas tecnologias se integram, de modo a suportar a arquitetura do software (protótipo).

4.2.1 Plataforma de Distribuição

A Torque Game Engine (TGE) é uma engine de jogos 3D, foi disponibilizada para comunidade de jogos Indie pela GarageGames, usada para fazer jogos multi-jogadores e jogos de um único jogador. É o produto de muitos anos de dedicação, trabalho, design interativo e desenvolvimento da equipe Dynamix em uma companhia de desenvolvimento de jogos muito conceituada e conhecida (MAURINA, 2003).

O Torque Engine (TORQUE, 2008) é um motor comercial multiplataforma (Windows, Linux e MacOS), possui editor de cenário em tempo-real, ou seja, durante o jogo podemos ir

adicionando recursos como objetos de cenário, itens, de maneira rápida e intuitiva, acelerando processo de desenvolvimento.

A TGE (Torque Game Engine) atualmente tem as seguintes características:

- **Multi-jogador:** TGE é baseada na arquitetura cliente-servidor suporta protocolos de comunicação para redes local e/ou Internet.
- **Raster-based¹¹:** A técnica de “*Raster-based graphics*”¹² é incorporada pela TGE.
- **Simulação de Eventos:** O design da TGE utiliza arquitetura cliente-servidor com loop¹³ de eventos que são aplicados para as lógicas dos jogos e das GUI por meio do sistema eventos.
- **Gerenciamento de Largura de banda e Memória:** Os requerimentos são baixos por conexão devido ao uso de algoritmos de agregação de mensagens nas transmissões de dados entre servidor e clientes.
- **Bibliotecas de Classes:** TGE possui com uma variedade de métodos e funções desenvolvida em C++ para implementações dos mundos virtuais onde processa operações de cálculos, ações e respostas.
- **Código Aberto:** Incorpora todos os códigos obtidos durante os muitos anos de desenvolvimento, fornecendo a possibilidade de implementação de novas bibliotecas pelo designer do software.
- **Integração:** Incluindo as renderização, captura todos os elementos, GUI, som e uso de técnicas de Realidade Virtual Distribuída.

¹¹ Em computação gráfica, uma imagem ou imagens raster bitmap é uma estrutura de dados que representam um modo geral rectangular, grade de pixels, ou pontos de cor, visualizáveis por meio de um monitor, papel ou por meio de visor.

¹² Um conjunto de instruções de software que é utilizado principalmente pelos recursos gráficos para desempenhar renderização de efeitos.

¹³ Execução contínua de instruções confirme a definição de uma condição.

A ferramenta utiliza a linguagem de script similar a C++, chamada de *TorqueScript* podendo ser atualizada em tempo real à produção do jogo. É ideal para o desenvolvimento de aplicações de curto prazo.

No TGE existem alguns termos referentes ao desenvolvimento de jogos por meio da engine como: interior, shape, datablock, IFL e imagens. Algumas dessas palavras têm significados específicos.

Os interiores são modelos criados por meio da classe *InteriorInstance*, frequentemente referenciada para simplificar a construção de interiores, é usada para exibir modelos que representam um objeto estrutural, no qual poderão ser incluído outros objetos tais como: prédios, pontes, muros e outras tipos de grandes estruturas. A técnica de modelagem de interiores é também usada para resolver alguns temas técnicos associados à criação de grandes e geometricamente complexos modelos que são contornados com uso de algoritmos de visibility culling, técnicas de detecção, de colisão e de iluminação. A técnica BSP denominada Binary Space Partitioning, usadas para detectar colisões contra objetos interiores (MAURINA, 2003).

A shape, também conhecida como um objeto DTS¹⁴ ou Dynamix Threespace Shape, é um modelo criado usando um editor de polígonos (ou equivalente); são geralmente usados para representar não-estruturalmente entidades, tais como árvores jogadores e veículos. (MAURINA, 2003). Podem possuir:

- Skeletal animations.
- Múltiplas texturas.
- Texturas animadas (IFL).
- Animações visíveis.
- Múltiplos Levels of Detail.
- Componentes translúcido e/ou componentes transparentes.
- Múltiplas caixas de colisão.

¹⁴ DTS é um tipo de arquivo 3D para ser importado pela Torque Engine.

Todos os objetos do jogo são arquivos DTS importados para dentro do jogo, como árvores, os bancos, as peças, o tabuleiro e as peças.

Um datablock é um objeto que pode ser declarado no código fonte da engine (C++), ou no código do TorqueScript. Declarado com várias propriedades, pode ser usado como um padrão para criar objetos. A Figura 4-1 mostra um exemplo de código TorqueScript usando datablock, para configuração das luzes do ambiente.

Objeto
<pre> datablock LightningData(LightningStorm) { strikeTextures(0) = "data/environment/lightning1frame1"; strikeTextures(1) = "data/environment/lightning1frame2"; strikeTextures(2) = "data/environment/lightning1frame3"; //strikeSound = LightningHitSound; thunderSounds(0) = ThunderCrash1Sound; thunderSounds(1) = ThunderCrash2Sound; thunderSounds(2) = ThunderCrash3Sound; thunderSounds(3) = ThunderCrash4Sound; }; </pre>

Figura 4-1: Configuração das luzes do ambiente virtual.

Os *datablocks* são especiais dentro da arquitetura da Torque pelas seguintes razão:

- Todos os *datablocks* são duplicados pelo servidor para cada cliente.
- Um *datablock* XYZ no servidor com ID 123 terá o mesmo nome XYZ e ID 123 em todos os clientes.
- *Datablocks* são transmitidos para clientes no início do jogo e não são atualizados depois disto, mantendo eles em efeito estático.
- *Datablocks* tem propriedades específicas de scripting.
- Por causa do conteúdo do datablock, é controlado pelo servidor e não pelo cliente, uma maneira eficiente de prevenir trapaças.

Os *datablocks* são blocos de informação, contendo modelo, textura, animações, entre outros parâmetros, que podem gerar diversos conteúdos repetidos. Um exemplo, dentro de um jogo ABC (um jogo multiplayer), os jogadores são permitidos a criar uma variedade de veículos com rodas. Cada um desses veículos possuem entre 4 a 8 pneus, texturas, efeitos

especiais (sons, emissores de partículas, etc), e uma longa lista de atributos físicos, um total aproximado de 2048 bytes, ou 2KB (MAURINA, 2003).

Foram usados datablocks para cada uma das peças, cilindros invisíveis, área do jogador e no cilindro de seleção. Estes são objetos únicos que são criados dinamicamente a cada início de jogo. No entanto para as imagens do jogo foi utilizado a IFL, que significa Image File Lists ou uma biblioteca para arquivos de imagens, durante o processamento de cada frame. Usar a biblioteca é uma alternativa viável para animação por meio de texturas, um exemplo e a criação de efeito para movimentação das peças por meio do IFL (Figura 4-2).



Figura 4-2: Textura do cilindro de seleção

A *engine* faz a troca da textura atual, por outra textura em um intervalo fixo de tempo. A animação é definida dentro de um arquivo de texto, onde é armazenado os nomes das texturas, seguidos pelos números de *frames* que ela irá se mostrar.

A textura da área do jogador adversário foi utilizada os IFLs, por meio de textura, devido a sua complexidade geométrica. Abaixo segue as imagens utilizadas para textura, bem como o conteúdo do arquivo IFL e o resultado final dentro do jogo. Exemplo de texturas de uma lista de texturas (*brilho0.png*, *brilho1.png*, *brilho2.png*, *brilho3.png* e *brilho4.png*) que forma a área do jogador (Figura 4-3).



Figura 4-3: Área do Jogador adversário.

Além desses aspectos tecnológicos, existe a questão de construção e definição da arquitetura, pois a *engine* suporta a customização de implementações independente da arquitetura determinada ao qual enriquece o projeto, pela flexibilidade.

4.2.2 OpenGL API Gráfica

OpenGL é definida como "um programa de interface para *hardware* gráfico" (MANSSOUR, 2006). Na verdade, OpenGL é uma biblioteca de rotinas gráficas e de modelagem, bi (2D) e tridimensional (3D), extremamente portátil e rápida. Usando OpenGL é possível criar gráficos 3D com uma qualidade visual próxima de um *ray tracer* (MANSSOUR, 2006). Entretanto, a maior vantagem para sua utilização é a rapidez, uma vez que os algoritmos são cuidadosamente desenvolvidos e otimizados pela *Silicon Graphics, Inc.*, líder mundial em Computação Gráfica e Animação.

OpenGL não é uma linguagem de programação, é uma poderosa e sofisticada API (*Application Programming Interface*) para criação de aplicações gráficas 2D e 3D (MANSSOUR, 2006). Seu funcionamento é semelhante ao de uma biblioteca C, uma vez que fornece uma série de funcionalidades. Normalmente quando se diz que um programa é baseado em OpenGL ou é uma aplicação OpenGL significa dizer que ele é escrito em uma linguagem de programação que faz chamadas a uma ou mais bibliotecas OpenGL.

As aplicações OpenGL variam de ferramentas CAD a programas de modelagem usados para criar personagens para o cinema, por exemplo, um dinossauro. Além do desenho de primitivas gráficas¹⁵, tais como linhas e polígonos, OpenGL dá suporte à iluminação, colorização, mapeamento de textura, transparência, animação, entre muitos outros efeitos especiais. Atualmente, OpenGL é reconhecida e aceita como um padrão API para desenvolvimento de aplicações gráficas 3D em tempo real (MANSSOUR, 2006).

Quando se está utilizando OpenGL, ao invés de descrever cena e como essa deve aparecer, basta determinar os passos necessários para se alcançar a aparência ou efeito desejado. Por meio, da GLUT¹⁶ é possível por meio de funções o gerenciamento de janelas, interação com o usuário ou arquivos de entrada/saída. Cada ambiente como, por exemplo, o *Microsoft Windows*, possui suas próprias funções para esses propósitos. OpenGL fornece um pequeno conjunto de primitivas gráficas para construção de modelos, tais como pontos, linhas e polígonos. A biblioteca GLU (que faz parte da implementação OpenGL) fornece várias

¹⁵ As estruturas e formas das quais derivam muitas outras formas de gráficos, ou seja, a Linha, o Retângulo, o Círculo.

¹⁶ O GLUT é uma ferramenta utilitária do OpenGL para desenvolvimento de interfaces do Windows.

funções para modelagem, tais como superfícies quádricas e curvas e superfícies NURBS (*Non Uniform Rational B-Splines*) (WRIGHT, 2000).

A palavra *pipeline* é usada para descrever um processo que pode ter dois ou mais passos distintos. A Figura 4.4 mostra uma versão simplificada do *pipeline* OpenGL. Como uma aplicação faz chamadas às funções API OpenGL, os comandos são colocados em um *buffer* de comandos. Este *buffer* é preenchido com comandos, vértices, dados de textura, etc. Quando este *buffer* é "esvaziado", os comandos e dados vão para o próximo estágio.

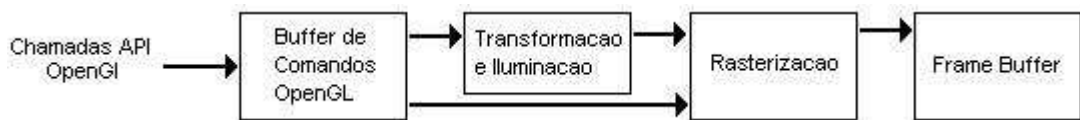


Figura 4-4: Versão simplificada do *pipeline* OpenGL (WRIGHT, 2000).

Após a etapa de aplicação das transformações geométricas e da iluminação, é feita a rasterização, isto é, é gerada a imagem a partir dos dados geométricos, cor e textura. A imagem final, então, é colocada no *frame buffer*, que é a memória do dispositivo gráfico. Isto significa que a imagem será exibida no monitor (WRIGHT, 2000).

A Torque Engine utiliza a API OpenGL para o desenvolvimento dos objetos e a faz renderizá-los, importando suas bibliotecas pela engine.

4.2.3 *Socket*

Segundo (SUN, 1999), a palavra *socket*, no contexto de redes de computadores, quer dizer um *ponto de comunicação* entre dois programas executando em uma rede. Esse *ponto* tanto envia quanto recebe dados, e possui uma *porta*, um número inteiro, para que a camada do protocolo TCP saiba para qual aplicação entregar os dados recebidos. Cada programa possui seu *socket* que o liga ao outro (Figura 4-5).

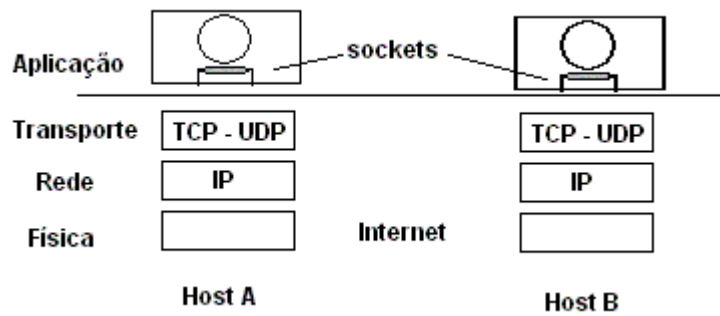


Figura 4-5: Modelo representa a comunicação de sockets e a pilha TCP/IP.

O *socket* permite então a comunicação processo a processo da seguinte forma:

- 1) Comunicação local: processos locais usando *sockets* locais
- 2) Comunicação remota: processos remotos usando *sockets* em rede (TCP/IP)

E possui dois tipos de serviço de transporte:

- 1) Datagrama - transporte não orientado a conexão e sem controle de erros (protocolo UDP)
- 2) *DataStream* - transporte orientado a conexão com controle de erros (protocolo TCP)

No trabalho foi adotado *socket*, devido à análise com outras tecnologias de distribuição como: CORBA, RMI e RPC. A partir de comparações (ATTROT, 2002) foram apontadas vantagens e desvantagens de cada um, por meio dos seguintes critérios: tempo de execução de chamadas a funcionalidades específicas, largura de banda alcançada e capacidade de suportar chamadas sucessivas de uma mesma funcionalidade.

No critério tempo de execução de chamadas a funcionalidades específicas conforme Attrot (2002), nas medições de tempo, o *socket* mostrou-se a mais rápida entre as quatro tecnologias. A média de tempo gasto por requisição de operação foi de 0,0027 ms. Com um mecanismo mais simples, mais sem a necessidade de serviços de nomes ou middleware, e usando chamadas diretas ao sistema operacional, saiu-se melhor em tempo de execução, contra o RMI de 0,37 ms/requisição, CORBA de 0,85 ms/requisição e RPC de 1,5 ms/requisição.

Foi analisada à largura de banda, em bytes/segundos, que cada tecnologia atinge quando transmite os dados (ATTROT, 2002). O *socket* alcançou 12222,2 bytes/segundos, o melhor resultados em comparação as outras. Entretanto no critério capacidade de suportar chamadas sucessivas de uma mesma funcionalidade, o *CORBA* e o *RMI* não suporta implementações concorrente ao contrario do *Socket* e *RPC*.

4.3. Arquitetura do Sistema

O diagrama ilustrado na Figura 4-6 apresenta uma visão geral do funcionamento e relacionamento entre as partes que compõem o sistema protótipo. Indica a interação entre jogador e o software proposto.

A arquitetura proposta é composta pelos seguintes módulos e sub-módulos:

1. GUI – Interface Gráfica com o Usuário;
2. Distribuidor de Eventos;
3. Módulo de Jogo Virtual dividido em dois sub-módulos internos: Objetos Virtuais e Algoritmos de Jogadas.

O sistema proposto utiliza técnicas de RV não-Imersivas: a tecnologia que suporta a interação com o usuário fundamenta-se por meio do mouse, monitor de vídeo e do teclado.

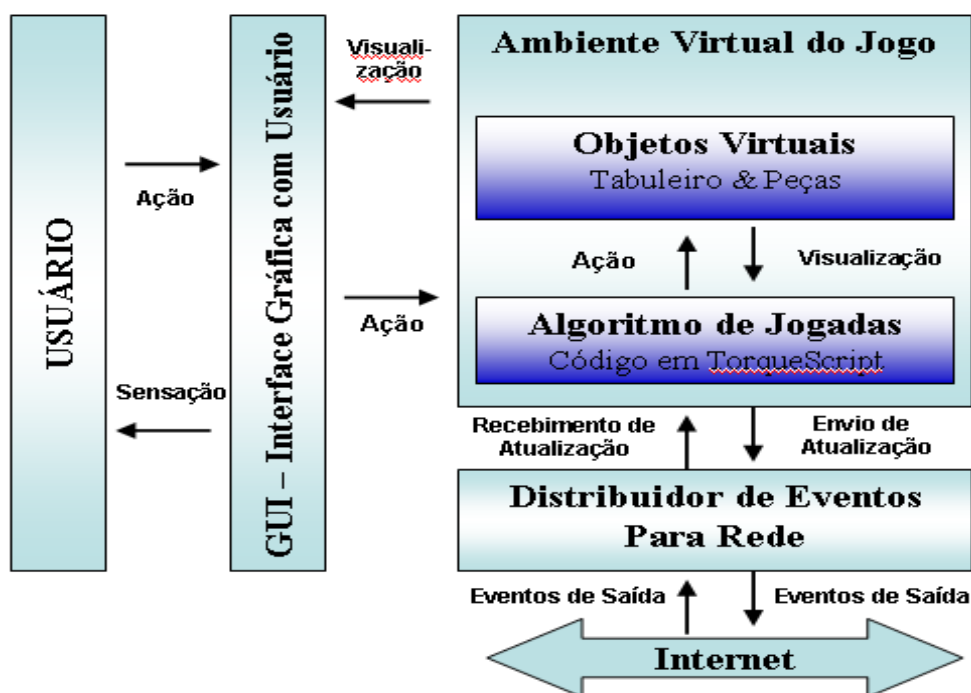


Figura 4-6: Arquitetura do software proposto Quarto 3D

A arquitetura é baseada em três componentes principais (Distribuidor de Eventos, GUI, Módulo Jogo Virtual), que formam o jogo Quarto 3D com a rede mundial de computadores, ou seja, Internet a aplicação proporciona a colaboração entre usuário conforme proposto.

4.3.1 GUI – Interface Gráfica com o Usuário

O módulo GUI disponibiliza o cenário do jogo e as opções de ajuda, configuração e execução do jogo. Entretanto, o cenário é composto por objetos previamente definidos para a visualização, navegação e interação. Por exemplo: quando o usuário realiza uma ação, recebe da GUI, a saída de informações, por meio de atualizações que proporcionam maior sensação de interação com o ambiente virtual (Figura 4-7).



Figura 4-7: Tela principal de apresentação do jogo.

4.3.2 Distribuidor de Eventos para Rede

O Distribuidor de Eventos foi implementado em TorqueScript para gerenciar as múltiplas conexões simultaneamente de envio de mensagens entre cliente e servidor. Quando se cria um jogo, é estabelecida uma aplicação servidora em modo de escuta; assim o próximo

jogador que autentica ao servidor, portanto abrirá uma conexão *socket TCP*, onde esse processo irá estabelecer a comunicação.

Por meio do método *accept()* para conecta-lo ao servidor, e suportar concorrentes conexões simplesmente com o processo de alocar threads, que irá monitorar por meio da porta aberta ao cliente pelo servidor cada cliente conectado. Porém o servidor (hosts) deverá ter um modo passivo para o fluxo de *socket*, por meio de uma porta de comunicação (5010), onde ficará esperando pelas atualizações das aplicações clientes e enviar (atualizados os clientes).

As atualizações dos usuários autenticados ao servidor serão por meio, de uma variável que contém todas as informações usadas para troca de mensagens com o cliente e/o servidor.

O Distribuidor de Eventos será responsável por enviar uma mensagem de atualização para todos os clientes autenticados, devido aos movimentos executados pelos jogadores. Após receber um evento de atualização vindo do servidor ao jogo do cliente, a GUI é atualizada exibindo ao usuário o último estado do jogo.

Na camada inferior do modelo, considerou-se a existência de protocolos de transporte. Apesar de ser importante numa comunicação de dados, esse estudo não objetiva detalhar essa camada.

4.3.3 Módulo de Jogo Virtual

Este módulo caracteriza a especificação dos objetos virtuais e algoritmos de jogadas, conforme descritos na arquitetura proposta (Figura 4-6).

4.3.3.1 Objetos Virtuais

Este sub-módulo representa o ambiente para jogar. A base do Ambiente 3D é um tabuleiro de dezesseis (16) peças em um jardim em que há reserva de espaço para o adversário.

Os objetos do jogo foram modelados por um software de modelagem 3Ds (3D Studio Max) e convertido por meio de um *loader* criando um arquivo para ser carregado pela Torque Game Engine para a montagem do Ambiente Virtual.

Para facilitar a comunicação entre os jogadores pensou-se numa interface que possibilitasse uma melhor visualização de todas as opções oferecidas pelo software. Além de agregar uma ferramenta que motiva a comunicação, que foi a criação de um chat é um visor de mensagens.

Outra mudança do *design* é permitir ao usuário, o recurso de caminhar pelo ambiente virtual, além de opções de câmeras para incentivar a concentração por meio de vários ângulos de visões.

A Figura 4-8 mostra a tela principal de apresentação do jogo. Nessa tela está disponível o cenário do jogo e cinco botões: o botão jogar, o botão opções, o botão ajuda, o botão créditos e o botão sair. Tais botões serão descritos com detalhes no tópico 4.4.4.

Entretanto, com a Figura 4-8 demonstra o resultado final do novo design do sistema proposto, ou seja, preparado para iniciar uma partida contra outro adversário.



Figura 4-8: Quarto3D – ambiente para jogar.

Outro aspecto do sub-módulo “objetos virtuais” e refere-se à movimentação das peças. Isto é, uma vez situada a peça em uma casa do tabuleiro, ela não pode mais ser removida até ao final da partida, assim os parâmetros são passados ao sub-modulo “algoritmos de jogadas” para processamento.

Com relação a essa movimentação, é importante ressaltar que, caso o usuário solte a peça fora do tabuleiro ou fora do suporte, ela voltará para a posição anterior.

4.3.3.2 Algoritmos de Jogadas

Este sub-módulo é responsável pela verificação das jogadas. Descreve o comportamento dos objetos virtuais por meio de equações matemáticas. Tais equações são implementadas a partir da linguagem *TorqueScript*.¹⁷

Ao realizar uma jogada, o usuário efetua a entrada de dados. Os dados são transmitidos para atualizar as variáveis de acordo com as equações matemáticas. Por exemplo: quando o usuário solta uma peça numa casa do tabuleiro, um código em *TorqueScript* realiza uma checagem em todas as linhas do tabuleiro (horizontal, vertical, diagonal) e analisa o conjunto de atributos de cada peça. Feita a verificação, ocorrem mudanças de comportamento no ambiente virtual; tais mudanças são feitas a partir de cálculos matemáticos. Neste caso, poderão ocorrer três situações:

- 1) não houve vencedor e ainda há peças para jogar.
- 2) acabaram as peças e não houve vencedor.
- 3) houve vencedor.

Se não houve vencedor e ainda há peça(s) para jogar (caso 1), uma mensagem indica o próximo jogador. O jogo continua normalmente. No caso de terem acabado as peças e de não haver vencedor (caso 2), o sistema acusa “Não houve vencedor”. Se houver um vencedor, o sistema questionará se o jogador deseja continuar.

Tais comportamentos serão detalhados no Capítulo 5 que tratará da implementação do sistema.

4.3.4 Estudo de Caso e Funcionamento do Sistema

Este módulo demonstra e justifica a escolha do estudo de caso, conforme descritos na motivação do trabalho.

¹⁷ A linguagem script da Torque engine.

4.3.4.1 Estudo de Caso “Ludos Top”

O jogo escolhido para integrar o ambiente proposto é o Quarto, premiado jogo francês criado por Blaise Müller em 1985 (FREITAS, 2008). É constituído de um tabuleiro 4 x 4 (16 casas) e de 16 peças. As peças (Figura 4-9) são representadas sem repetição em sólidos geométricos: paralelepípedos e cilindros que diferem a partir dos seguintes atributos: forma, tamanho, cor, existência ou não de um furo.

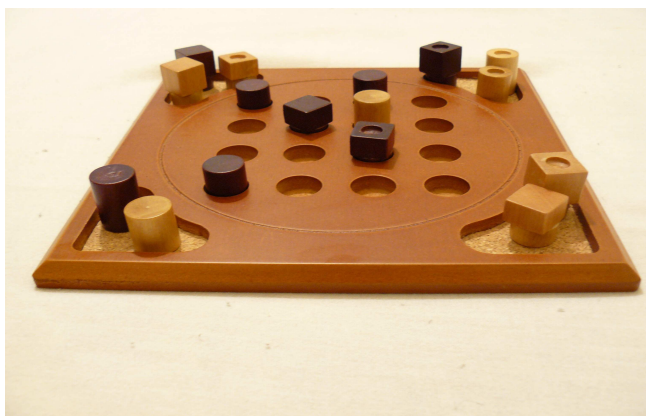


Figura 4-9: As 16 peças do jogo, mais 4 x 4 casa (16) no tabuleiro.

O objetivo do jogo é alinhar peças com atributos semelhantes. Vence a partida o jogador que conseguir alinhar peças que tenham um ou mais atributos comuns. Exemplo: se o jogador alinhar cilindros, com furo ou sem furo, de uma ou outra cor, sejam altos ou baixos, ele vence a partida porque conseguiu reunir peças com pelo menos um atributo em comum: a forma cilíndrica.¹⁸



Figura 4-10: Tabuleiro do jogo Quarto

¹⁸ O alinhamento pode ser tanto na vertical, na horizontal ou na diagonal. (Figura 4-2)

O jogo Quarto implica questões pedagógicas interessantes, razão pela qual foi premiado. Uma delas é a questão da socialização. Jogado em dupla, obriga o jogador a um diálogo com o adversário. Faz com que tenha de se colocar no lugar do outro, ponto de partida para as estratégias que terá de construir para vencer. A questão da socialização, a propósito, é um dos aspectos que caracterizam ao período operatório concreto (7 a 11 anos) (FREITAS, 2008).

Outro aspecto interessante é o estímulo à capacidade de observação. Esta habilidade se torna presente no momento de identificar conjuntos ou grupos, definidos a partir de características específicas (cor, tamanho, forma, existência ou não de um furo). Ao mesmo tempo, não se pode esquecer de que tais peças, formando subgrupos, pertencem a um grupo maior (o das Figuras geométricas).

Esta questão da classificação acontece no momento em que a criança seleciona as peças que vai jogar. Neste momento, ela começa a distinguir que, no universo das peças do jogo, há algumas que possuem características comuns (FREITAS, 2008).

É interessante dizer também que o software proposto poderá estimular o raciocínio não somente pelo fato da inclusão dos vários jogadores no jogo Quarto, mas também por estabelecer a interação entre aqueles que estão jogando, via Internet, pois em qualquer momento pode-se tentar alinhar peças com um ou mais atributo em comum e validar o alinhamento (FREITAS, 2008).

Outro aspecto interessante do jogo é seu caráter competitivo. Trata-se de um aspecto normalmente comum nos jogos infantis, sendo considerado saudável, se vivenciado de modo equilibrado. Torna-se um estímulo para criança, incentivando seu envolvimento com o conteúdo pedagógico do jogo (ABED, 2000) (MACEDO, 2000).

É importante destacar que Freitas (2008), implementou o jogo para a versão digital (Figura 4-11), atribuindo por meio da linguagem VRML em conjunto com o JavaScript.

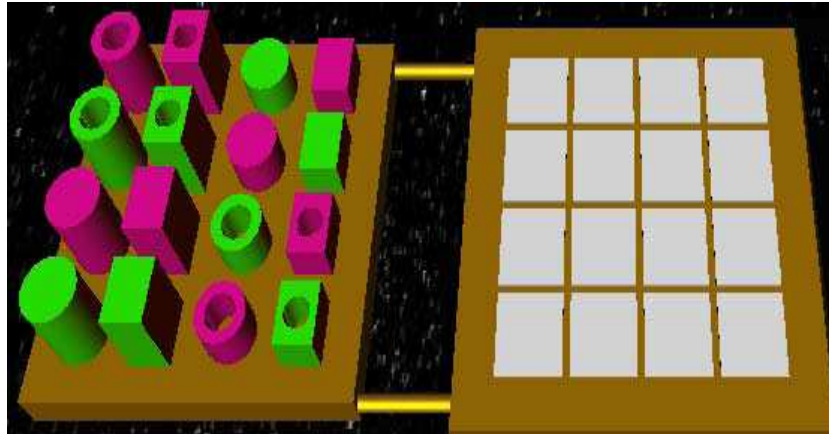


Figura 4-11: LudosTop na versão VRML (FREITAS, 2008).

Conforme avaliado no tópico 3.3.4, esse sistema possuía algumas limitações que motivaram como estudo de caso. As limitações desta versão do LudosTop são:

- 1) Ausência de uma ferramenta integrada de colaboração como chat.
- 2) Ser mono-usuário.
- 3) Não estava implementada todas as regras do jogo original (Por exemplo: a regra de quem ganhar o jogo quem observar o quarto primeiro).
- 4) Suporte a internet, com estrutura distribuída eficiente.
- 5) Não permitir que o professor participe ativamente das partidas com os jogadores.
- 6) Aplicar um software como uma ferramenta de EAD multi-jogador sobre a Internet e distribuído.

Esses foram os fatores que motivaram, o LudosTop como estudo de caso. É preciso destacar, ainda, que o desempenho do software proposto é um dos pontos focos do jogo, tornando a aplicação mais atraente.

4.3.4.2 Funcionamento do software proposto

Nesta seção, será descrito o funcionamento do sistema e sua organização.

Como já foi mencionado no tópico 4.4.3.1, a Figura 4-7, demonstra a interface gráfica do usuário. A GUI é composta pelos seguintes botões: Jogar, Opções: Ajuda, Crédito e Sair.

Ao selecionar o botão “Jogar”, o jogador irá fazer a autenticação (Figura 4-12), digitando um apelido e o nome de usuário. Isso identificará o usuário que está conectado e o apelido pelo qual será chamado durante o jogo. Após a autenticação, o jogador poderá criar uma partida ou verificar se algum adversário já está conectado.

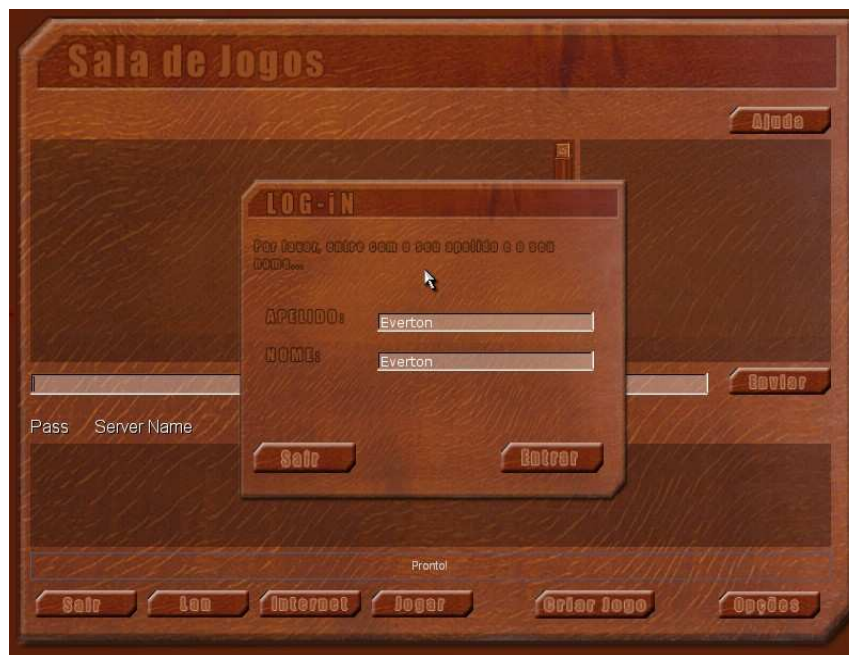


Figura 4-12: Tela de autenticação.

O botão “Opções” (Figura 4-13) disponibiliza opções para áudio e visualização do jogo. Na opção “Gráficos”, é possível selecionar o driver gráfico a ser usado, além da resolução e configuração de bit para exibição. Na opção “áudio”, é possível definir driver de som e volume.



Figura 4-13: Opção de áudio e visualização.

A idéia inicial foi elaborar um guia com as explicações sobre o jogo, permitindo maior interação, mais facilidade e agilidade ao executar o jogo pela primeira vez. O jogador terá informações sobre o jogo, regras e comandos. (Figura 4-14)



Figura 4-14: Tela ajuda sobre o jogo.

O botão “Créditos” conforme passado na Figura 4-15, uma interface com os nomes daqueles que desenvolveram o software proposto.



Figura 4-15: Informações dos desenvolvedores.

A Figura 4-16, mostra os botões “Lan” e “Internet” que verificará a existência de um jogo criado via LAN¹⁹ ou Internet, pois assim o jogador escolher seu adversário.



Figura 4-16: Criação de um jogo

Ao criar o jogo com botão Criar, o usuário deverá aguardar seu oponente, conectar conforme demonstrado na Figura 4-12. Assim começa uma partida, o jogador após ter se

¹⁹ Local Area Network

conectado ao jogo, poderá enviar mensagens de texto a todos os usuários conectados e assim selecionar uma peça para o oponente, conforme explicado no botão “Ajuda”.

O intuito da criação do chat, não só para o ensino à distância, mas como instrumento de construção de conhecimentos, pesquisa, troca de informação e comunicação entre os jogadores. No chat, os jogadores interagem com os outros participantes por meio da comunicação escrita, utilizando códigos peculiares do ambiente virtual. Essa ferramenta pode contribuir para o aprimoramento da capacidade de raciocínio e agilidade na escrita.

Em qualquer momento, o usuário poderá pressionar o botão “Q” na tela e validar o alinhamento do Quarto (Figura 4-17).



Figura 4-17: Tela do vencedor da partida.

Portanto, o ato de pensar e fazer educação assume um novo paradigma onde, na era digital, o aluno passa a ser o gerenciador do seu próprio conhecimento, sendo o professor o mediador nessa construção do saber. Podemos perceber que o chat é uma ferramenta que favorece a comunicação entre grupos. Assim software proposto tem, possui o recurso de permitir que o professor também se conecte ao jogo, podendo interagir também por meio do chat. Esta ferramenta pode ser uma solução para os problemas crônicos do processo educacional brasileiro.

No Jogo Quarto 3D, uma vez conseguido fazer o alinhamento é perguntado aos jogadores, se deseja iniciar outro jogo.

4.3.5 Hardware requerido para o funcionamento do sistema

Para que o sistema possa operar, requer:

- Sistema Operacional: Microsoft, WindowsME/2000/XP;
- Processador: Pentium II 400 MHz ou superior;
- Memória RAM: mínimo de 128 MB;
- Espaço livre em disco: 500 MB livre para armazenar arquivos de programa;
- Resolução: 600 x 800;
- Placa de vídeo: AGP Radeon ou GeForce class w/32MB;
- Dispositivos: Mouse, Video e teclado;
- Acesso à Internet via telefone ou banda larga;
- Placa de som: opcional.

4.4. Considerações Finais

Nessas considerações finais é importante mencionar que a tecnologia utilizada (linguagens *TorqueScript* e *OpenGL API*) visou criar um software bastante acessível tanto do ponto de vista operacional quanto aquisitivo.²⁰

O objetivo era, utilizar técnicas de Realidade Virtual Distribuída que, integradas a um estudo de caso consistente, que permitisse a elaboração de um software educacional interativo, gratuito na Internet e adequado à faixa etária envolvida.

No próximo capítulo, serão descritos os detalhes da implementação do sistema proposto.

²⁰ Do ponto de vista aquisitivo, parte-se do princípio de que as escolas já estejam munidas de computadores e conectadas à internet. Essa é uma tendência, aliás, apontada e apoiada financeiramente pelo governo federal.

Capítulo 5

5. Detalhes de Implementação do Jogo

5.1. Introdução

Este capítulo apresenta detalhes de implementação do Jogo Virtual, cuja arquitetura foi descrita no Capítulo 4.

A implementação deste módulo foi feita utilizando-se a Linguagem C++, OpenGL API e *TorqueScript*.

5.2. Detalhes de Implementação

Para modelagem geométrica, usou-se 3D Studio Max. A modelagem foi feita, utilizando-se da técnica denominada “Box Modelling”, que consiste em modelar objetos a partir de peças básicas como caixotes, cilindros ou qualquer outra forma básica.

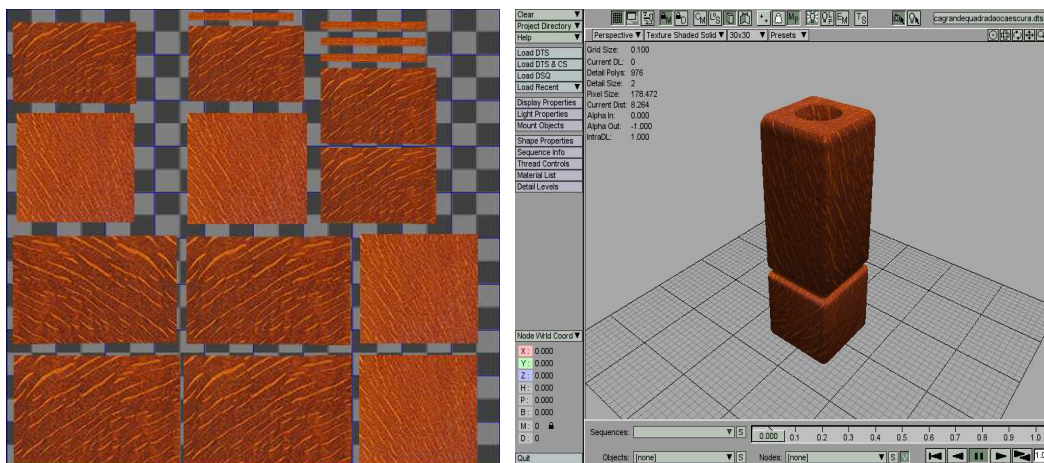


Figura 5-1: Modelagem das peças do jogo Quarto3D.

Ao encontrar a forma adequada, o objeto foi aberto utilizando um mapa UVW²¹ para que pudesse ser texturizado. Isso deve ser feito em cada um dos modelos, para que não haja uma textura esticada ou defeituosa.

Cumprida essa primeira etapa, será criado o cenário na Torque Game Engine. Os elementos são importados e o cenário pode ser montado em tempo real.

A partir do momento em que os modelos 3D das peças e dos elementos são importados pela *Torque Engine*, o ambiente é construído. O ambiente se baseia no modelo preparado, neste caso um parque (Figura 5-2) composto de vegetação rasteira e árvores. Além disso, uma mesa com tabuleiro e dois assentos. As peças foram posicionadas nas bordas do tabuleiro, de forma a facilitar a visualização dos jogadores.

O módulo Jogo Virtual está dividido em dois sub-módulos: Objetos Virtuais e Algoritmos de Jogadas que interagem com o Distribuidor de Eventos.

5.2.1 Objetos Virtuais

O cenário foi montado com a ajuda do *World Editor*, uma ferramenta de edição de cenário em tempo real da própria TGE. Com ele, foi possível criar um cenário complexo, com extrema precisão e facilidade, adicionando diversos recursos, como luzes, sons 3D e triggers.



Figura 5-2: Modelagem do cenário do Jogo

²¹ O UVW Map: mapeia o modelo para poder aplicar a textura com precisão. (Maurina, 2003)

Uma vez posicionados os objetos, e salva a missão, o Torque gera um arquivo denominado Mission, contendo a descrição de todos os elementos a ser carregados no cenário.

A Figura 5.3, mostra um trecho das implementações, com os códigos computacionais em que podem ser vistos os comandos relativos aos objetos criados com *World Editor* para formação do ambiente.

```

/* quarto.contents/data/missions/quarto.mis */

/*...*/
new SimGroup(MissionGroup) {
    canSaveDynamicFields = "1";

    new ScriptObject(MissionInfo) {
        desc0 = "Cenario do Quarto";
        descLines = "1";
        name = "Quarto";
    };
    new MissionArea(MissionArea) {
        canSaveDynamicFields = "1";
        Area = "-360 -648 720 1296";
        flightCeiling = "300";
        flightCeilingRange = "20";
        locked = "true";
    };
    new TerrainBlock(terrain) {
        canSaveDynamicFields = "1";
        rotation = "1 0 0 0";
        scale = "1 1 1";
        terrainFile = "./quarto.ter";
        squareSize = "8";
        bumpScale = "2";
        bumpOffset = "0.007";
        zeroBumpScale = "7";
        tile = "1";    position = "-1024 -1024 0";
        locked = "true";
    };
/*...*/

```

Figura 5-3: Código de implementação do ambiente virtual do jogo

Uma vez posicionados os objetos, fez-se necessário distribuir alguns para mapear o tabuleiro (Figura 5-4), uma vez que o objeto em si não possui essa facilidade. Para isso, foram dispostos diversos objetos invisíveis, com os quais, ao soltar a peça sobre o tabuleiro, as peças transparentes seriam responsáveis por verificar a colisão com o mouse, efetuando assim a jogada. Eles também têm por objetivo informar a posição inicial das peças e a área de destino das peças.

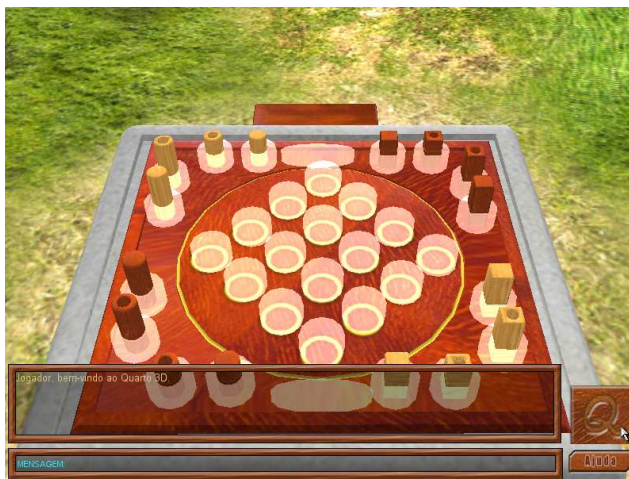


Figura 5-4: Posições dos objetos invisíveis.

A câmera (Figura 5-4) foi posicionada a partir de um ponto central no meio do tabuleiro. Esse ponto é usado como personagem de jogo, desse modo se pode contornar a câmera por todo o cenário, controlando a angulação da câmera e o seu campo de visão. (*Field of View - FOV*).

As peças do jogo são declaradas como *StaticShapeData* (Figura 5-5), um tipo mais evoluído dentre os objetos encontrados na Torque Game Engine. Esse deriva do objeto *TSSStatic*, que nada mais é que um simples objeto 3D, como é o caso dos elementos do cenário.

```
datablock StaticShapeData(QSSPecaGrandeQuadradaClara)
{
    observeThroughObject = true;
    category = "QSShapes"; // Quarto Static Shapes
    className = "QSPieces";
    shapeFile =
    "~/data/shapes/gardenobjects/pecas/pecagrandequadradaclara.dts";
    isInvincible = true;
};
```

Figura 5-5: Exemplo de codificação para carregar objetos do 3D Max.



Figura 5-6: Possíveis câmeras do software proposto.

A Figura 5-6 demonstra alguns pontos de vista possíveis para o usuário durante o transcurso de uma partida online.

5.2.2 Algoritmos de Jogadas

Este sub-módulo é responsável pela verificação das jogadas. Descreve o comportamento dos objetos virtuais por meio de fórmulas matemáticas e envia atualizações ao distribuidor de eventos que atualiza os clientes conectados ao servidor. Tais operações são implementadas a partir do *TorqueScript*.

As peças se movimentam de acordo com uma função de projeção realizada pela API *OpenGL*, usando a função *gluUnProject* que converte um ponto da tela para o cenário, a partir de algum ponto que seria o observador, no caso a câmera. Exemplo da definição da função *gluUnProject*, com os parâmetros a ser passados para execução.

```
GLint gluUnProject( GLdouble winX, GLdouble winY, GLdouble winZ, const GLdouble
*model, const GLdouble *proj, const GLint *view, GLdouble*  objX, GLdouble*  objY,
GLdouble*  objZ )
```

Na Figura 5-7, mostrando o código fonte da torque enfocando a função para executar o movimento de uma peça.

```
// Em engine/gui/core/GuiTSCtrl.cc
// Utilização da gluUnProject dentro da Engine TGE
/*...*/
bool GuiTSCtrl::unproject(const Point3F &pt, Point3F *dest)
{
    GLdouble objx, objy, objz;
    GLint result = gluUnProject(pt.x, pt.y, pt.z,
                               mSaveModelview, mSaveProjection, mSaveViewport,
                               &objx, &objy, &objz);
    if(result == GL_FALSE)
        return false;
    dest->set(objx, objy, objz);
    return true;
}
/*...*/
```

Figura 5-7: Implementação na biblioteca da Torque em C++.

O funcionamento das jogadas do Quarto começa com o jogador entregando ao adversário uma peça. Para fazer esse movimento é necessário que primeiramente ele selecione uma jogada.

Os arquivos que compõem esses algoritmos são *quarto.contents/client/scripts/playGui.cs* e *quarto.contents.server/scripts/commands.cs*. Esses arquivos contêm toda a parte de logística de movimentação do mouse.

O objeto PlayGui (Figura 5-8) é o responsável por capturar as ações do mouse (Mouse Down, Mouse Up, Mouse Move (Figura 5-9) e Mouse Dragged). Cada um desses eventos possui uma aplicação no jogo.

```
// Em engine/game/GameTSCtrl.h
/*...*/
private:
    Point3F mMouse3DVec;
    Point3F mMouse3DPos;
    Point3F mObjReferencePos;

public:
    void onMouseMove(const GuiEvent &evt);
    void onMouseDown(const GuiEvent &evt);
    void onMouseUp(const GuiEvent &evt);
    void onMouseDragged(const GuiEvent &evt);
    Point3F getMouse3DVec() {return mMouse3DVec;};
    Point3F getMouse3DPos() {return mMouse3DPos;};
/*...*/
```

Figura 5-8: Codificação de eventos do mouse em C++.

A `PlayGui` é uma instância do objeto `GameTSCtrl` (Figura 5-9). Esta classe inicialmente não possui os eventos de mouse implementados. Por essa razão foi necessário fazer a implementação destas funções para que pudéssemos utilizar o Mouse 3D.

```
// Em engine/game/GameTSCtrl.cc
// Demonstração do onMouseMove.
void GameTSCtrl::onMouseMove(const GuiEvent &evt)
{
    if(gSnapLine)
        return;

    MatrixF mat;
    Point3F vel;
    if ( GameGetCameraTransform(&mat, &vel) )
    {
        Point3F pos;// = mObjReferencePos;
        mat.getColumn(3,&pos);
        const Point3F screenPoint(evt.mousePosition.x, evt.mousePosition.y, 1);
        Point3F worldPoint;

        if (unproject(screenPoint, &worldPoint))
        {
            Point3F vec = worldPoint - pos;

            lineTestStart = pos;
            vec.normalizeSafe();
            mMouse3DPos = pos;
            mMouse3DVec = vec;
            // Função que leva para a PlayGui::onMouseMove
            Con::executef(this, 2, "onMouseMove");
        }
    }
}
```

Figura 5-9: Implementação da função `onMouseMove()`.

Na Figura 5-10, apresenta a função `PlayGui::onMouseDown(%this)` armazena a posição 3D do mouse e a sua direção a partir da câmera. Ao fazer isso, ele envia um comando ao Servidor, contendo essas informações.

```
commandToServer('selectObject', %mouseVec, %cameraPoint);
```

Esse comando tem como função dizer ao servidor que o botão do mouse foi clicado, e se for durante sua própria jogada, ele poderá evoluir o `onMouseDown` para `onMouseDownDragged` e arrastar uma peça para que seja jogada no tabuleiro por meio da função `onMouseDownUp`.

Entretanto, na Figura 5-10 descreve a implementação do código que refere-se ao processo de seleção e movimentação das peças.

```

// Em PlayGui.cs
function PlayGui::onMouseDown(%this)
{
    // Recupera o vetor direção em relação a câmera e a posição 3D do mouse.
    %mouseVec = %this.getMouse3DVec();
    // Posição 3D da câmera.
    %cameraPoint = %this.getMouse3DPos();
    // Mensagem enviada ao servidor dizer para selecionar um Objeto.
    commandToServer('selectObject', %mouseVec, %cameraPoint);
}

// Em Commands.cs
function serverCmdSelectObject(%client, %mouseVec, %cameraPoint)
{
    /*...*/
    %selectRange = 200;
    %mouseScaled = VectorScale(%mouseVec, %selectRange);
    %rangeEnd = VectorAdd(%cameraPoint, %mouseScaled);
    %searchMasks = $TypeMasks::StaticShapeObjectType;
    %scanTarg = ContainerRayCast (%cameraPoint, %rangeEnd, %searchMasks,
false);
    /*...*/
}

```

Figura 5-10: Exemplificando a execução da função pela Torque.

Ao ativar o evento *onMouseDown* da *PlayGui*, o mouse envia por meio do comando *commandToServer*, e o jogador consegue “empacotar” os dados com relação à posição do mouse do cliente, e sua câmera. A partir disso, em *serverCmdSelectObject*, é definido o range, ou seja, uma distância de até onde será traçado o *ContainerRayCast*, uma função que recupera o primeiro objeto com as características enviadas, no nosso caso *\$TypeMasks::StaticShapeObjectType*.

A seguir, ele verifica qual o estado em que o turno está e dependendo de cada um desses estados ele processará de uma maneira diferente. Por exemplo, se for o estado *\$ESTADO_ESCOLHER_PECA* ele irá criar uma área com um objeto que utiliza uma IFL para fazer animações, informando para onde deve ser levada a peça.

A peça seguirá o mouse por meio do evento *onMouseDragged* da *PlayGui* e depois será colocada no lugar onde foi deixada ao realizar o *onMouseUp*, após uma prévia certificação.

O botão Quarto (Q) foi implementado para que qualquer um dos dois jogadores possa apertá-lo em qualquer momento do jogo.²² Uma vez apertado, este jogador irá para o estado \$ESTADO_SELECIONAR_QUARTO, enquanto o outro vai para o \$ESTADO_AGUARDAR_QUARTO.

Cada peça selecionada é armazenada em uma variável do tipo Array. Após o Jogador terminar de selecionar as peças que ele julga formar Quarto, imediatamente o servidor executa a função (Figura 5-11) *verificaQuarto()*. Essa função tem como objetivo, conferir primeiramente se as peças selecionadas estão em posição de formar um Quarto.

```
// Em Quarto.cs
function verificaQuarto()
{
    /*...*/
    // Verifica se a linha é valida.
    if(!((%ind(0) == 1 && %ind(1) == 2 && %ind(2) == 3 && %ind(3) == 4)
    || (%ind(0) == 5 && %ind(1) == 6 && %ind(2) == 7 && %ind(3) == 8)
    || (%ind(0) == 9 && %ind(1) == 10 && %ind(2) == 11 && %ind(3) == 12)
    || (%ind(0) == 13 && %ind(1) == 14 && %ind(2) == 15 && %ind(3) == 16)
    || (%ind(0) == 1 && %ind(1) == 5 && %ind(2) == 9 && %ind(3) == 13)
    || (%ind(0) == 2 && %ind(1) == 6 && %ind(2) == 10 && %ind(3) == 14)
    || (%ind(0) == 3 && %ind(1) == 7 && %ind(2) == 11 && %ind(3) == 15)
    || (%ind(0) == 4 && %ind(1) == 8 && %ind(2) == 12 && %ind(3) == 16)
    || (%ind(0) == 1 && %ind(1) == 6 && %ind(2) == 11 && %ind(3) == 16)
    || (%ind(0) == 4 && %ind(1) == 7 && %ind(2) == 10 && %ind(3) == 13)))
    {
        cancelaQuarto();
        return;
    }

    if(verificaPecas($Game::pecasQuarto(0), $Game::pecasQuarto(1),
    $Game::pecasQuarto(2), $Game::pecasQuarto(3)))
    {
        iluminaCilindros(%ind(0), %ind(1), %ind(2), %ind(3));
        // Fim de jogo
        fimDeJogo();
        return true;
    }
    else
    {
        return false;
    }
}
}
```

Figura 5-11: Algoritmo de execução das regras do jogo.

Caso ele não esteja em posição de formar Quarto, então cancela-se o estado, caso contrário ele faz a verificação das peças e se estiver correto, declara final de jogo conforme detalhado na Figura 5-12.

²² A qualquer momento do jogo, após 4 jogadas.

```
// Em Quarto.cs

function verificaPecas(%peca1, %peca2, %peca3, %peca4)
{
    // Se todas as peças existirem
    if(%peca1 != -1 && %peca2 != -1 && %peca3 != -1 && %peca4 != -1)
    {
        if(%peca1.grande == %peca2.grande == %peca3.grande == %peca4.grande)
            // Se elas forem grandes ou pequenas
            {
                return true;
            }
            else
        if(%peca1.redonda == %peca2.redonda == %peca3.redonda == %peca4.redonda)
            // Se elas forem redondas ou quadradas
            {
                return true;
            }
            else
        if(%peca1.oca == %peca2.oca == %peca3.oca == %peca4.oca)
            // Se elas forem ocas ou sólidas
            {
                return true;
            }
            else
        if(%peca1.escura == %peca2.escura == %peca3.escura == %peca4.escura)
            // Se elas forem escuras ou claras
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        else
        {
            return false;
        }
    }
}
```

Figura 5-12: Código de verificação do alinhamento do quarto.

5.2.3 Distribuidor de Eventos

Conforme mencionado a Torque possui uma arquitetura Cliente-Servidor (Capítulo 4.3.1). Alguns processos são processados unicamente no cliente, enquanto o jogo em si é processado apenas no servidor.

O Distribuidor de Eventos foi implementado para manipular o processo de comunicação entre clientes e servidores, onde atualizam os estados do Ambiente Virtual, devido a interação do usuário.

Para resolver este problema foram criadas duas funções na Torque:

```
commandToClient(%cliente, 'função', %param1, %param2, ...);

commandToServer('função', %param1, %param2, ...);
```

Quando alguma informação quer ser atualizada sobre um cliente, ou alguma ação que dependa do servidor, o cliente faz uma chamada em um *commandToServer*, passando o nome da função e os parâmetros pelos quais o servidor irá se encarregar e processar adequadamente, atualizando todo o jogo. Um exemplo da movimentação é demonstrada na Figura 5-13:

```
// Em PlayGui.cs

function PlayGui::onMouseMove(%this)
{
// mouseVec = vector from camera point to 3D to mouse coords
  %mouseVec = %this.getMouse3DVec();
  // cameraPoint = the world position of camera
  %cameraPoint = %this.getMouse3DPos();

  if(%this.isSelectingObj())
  {
      commandToServer('putObjectAtPlace', %mouseVec,
%cameraPoint);
  }
  else
  {
      commandToServer('focusObject', %mouseVec, %cameraPoint);
  }
}

//-----
// Em Commands.cs
// Coloca a peça se for um cilindro, senão volta para pos original
function serverCmdPutObjectAtPlace(%client, %mouseVec,
%cameraPoint)
{...}

// Muda a cor do objeto, fazendo o foco
function serverCmdFocusObject(%client, %mouseVec, %cameraPoint)
{...}
```

Figura 5-13: Codificação de movimentação das peças e câmera.

Do mesmo modo, são passadas informações para os clientes por meio da função *commandToClient*, onde é enviado a atualização a todos usuários autenticados conforme Figura 5-14.

```
// Em Quarto.cs

function mudaEstado(%client, %estado)
{
    commandToClient(%client, 'mudaEstado', %estado);
}

function clientCmdMudaEstado(%estado)
{
    clientCmdRemoveGuis();
    $client::estadoAntigo = $client::estado;
    $client::estado = %estado;

    carregaGui(%estado);
}
```

Figura 5-14: Função de atualização de estados do cliente.

O Distribuidor de Eventos tem a responsabilidade de gerenciar o mecanismos de distribuição, com a arquitetura cliente-servidor não apresentando grandes dificuldades para implementação, uma vez que é separado o papel de cada entidade durante o funcionamento do jogo.

5.3. Considerações Finais

Este capítulo apresentou a implementação, do software proposto em caráter protótipo. Como foi mostrado, tal sistema foi desenvolvido por meio de técnicas de Realidade Virtual não-imersiva, proporcionadas pela integração entre a Linguagem C++, OpenGL API e Torque Engine.

No próximo capítulo, será narrada uma experiência de utilização do software por crianças do Ensino Fundamental.

Capítulo 6

6. Resultado da Aplicação do Software

6.1. Introdução

Este capítulo descreve a aplicação do software em uma escola do Ensino Fundamental. O objetivo desta aplicação foi verificar o modo como estudantes e professores receberiam o programa, além de validar a interatividade proporcionada pela distribuição.

Após utilizar o software, as crianças ou estudantes responderam a um questionário sobre sua familiaridade com jogos de computador e videogames, sobre o interesse despertado pelo jogo Quarto 3D, bem como sobre a facilidade de manuseio do sistema.

Um experimento com o software foi feito na Escola Técnica Fortec de São Vicente. O jogo foi apresentado a 2 (dois) professores e a trinta e quatro (34) crianças de 5ª série do Ensino Fundamental. A faixa etária variou entre onze (11) e doze (12) anos. A escola forneceu as condições ideais de uso do software (uma criança por computador e computadores com acesso à Internet), assim como tempo suficiente para o envolvimento dos novos usuários com o software. O encontro começou com os professores. Depois de uma conversa sobre os objetivos do software, foi lhes dada a oportunidade de utilizar o programa e de expressar suas impressões. O próximo passo foi apresentá-lo às crianças.

6.2. Avaliação do Sistema pelos usuários

Em relação ao encontro com os professores, é importante assinalar que todos julgaram o software interessante e útil. Um dos usuários afirmou: “É muito interessante, pois demonstra um senso de concentração e observação para jogá-lo”.

Em relação aos alunos, foi feito um levantamento das respostas. A seguir, serão apresentados e comentados os dados resultantes desse levantamento.

1. Uso do computador

Conforme o questionário, a primeira indagação é sobre o uso dos computadores além de descobrir os principais motivos, se refere ao uso.

De acordo com a Figura 6-1, constata-se que a maioria dos alunos utiliza o computador para efetuar pesquisas e jogos. Essa afirmação pode ser feita devido ao apontamento das respostas na pesquisa, pois evidenciando que a prática de jogos tem se tornado parte do dia-a-dia das crianças. Além disso, os dados apresentam que o computador tem-se tornado cada vez mais um componente na vida das crianças.

Tabela 6-1: Dados referentes ao uso do computador pelos alunos pesquisados.

		Freqüência	%	% Válido
Válido	Para jogar	6	17,6	17,6
	Fazer pesquisas	2	5,9	5,9
	Para brincar	5	14,7	14,7
	Mais de uma opção	17	50,0	50,0
	Outros	4	11,8	11,8
	Total	34	100,0	
Em branco				
Total		34	100,0	100,0

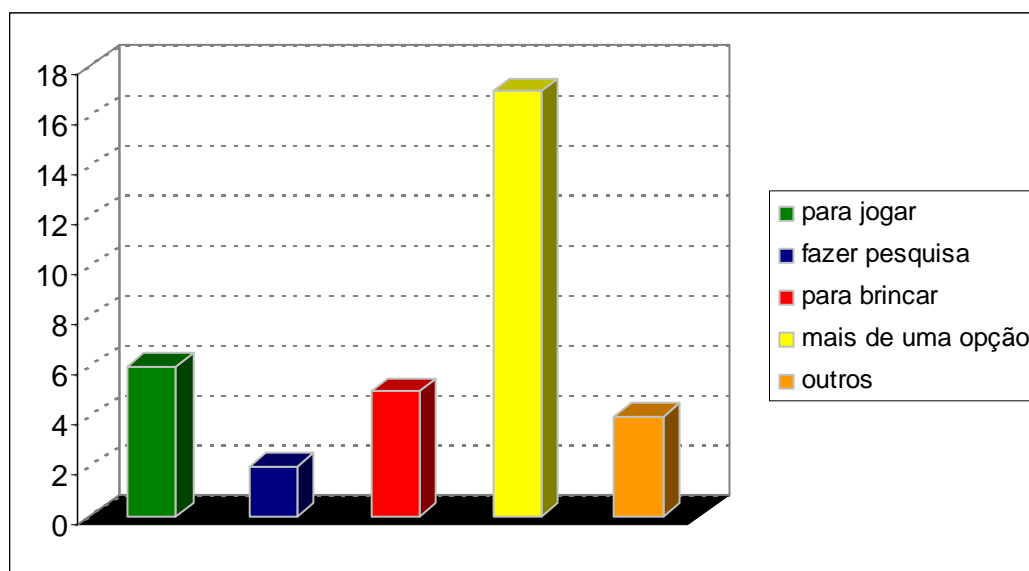


Figura 6-1: A demonstração gráfica sobre o uso do computador.

Com a tabela 6-1, foram registradas todas as reações e opiniões referentes ao uso do computador, e demonstra que 50% dos usuários fazem pesquisa, brinca e joga pelo

computador. Portanto, todos os participantes usam o computador como uma ferramenta de trabalho e diversão.

2. Avaliação Geral do software quanto à sua utilidade

Como mostra a Figura 6.2, o software despertou o interesse nos estudantes, quanto à aceitação, devido a sua interatividade e possibilidade de jogar pela Internet, foi um dos fatores, mas outra característica que motivou os estudantes ao uso do software propostos é a possibilidade de trocar mensagens enquanto está jogando, embora seja apenas um chat esse mecanismos de colaboração permitindo a experiência de fazer amigos.

Tabela 6-2: Dados referentes à questão: uso do software.

		Freqüência	%	% Válido
Válido	Fácil	27	79,5	79,5
	Médio	6	17,6	17,6
	Difícil	1	2,9	2,9
	Total	34		
Em branco				
Total		34	100,0	100,00

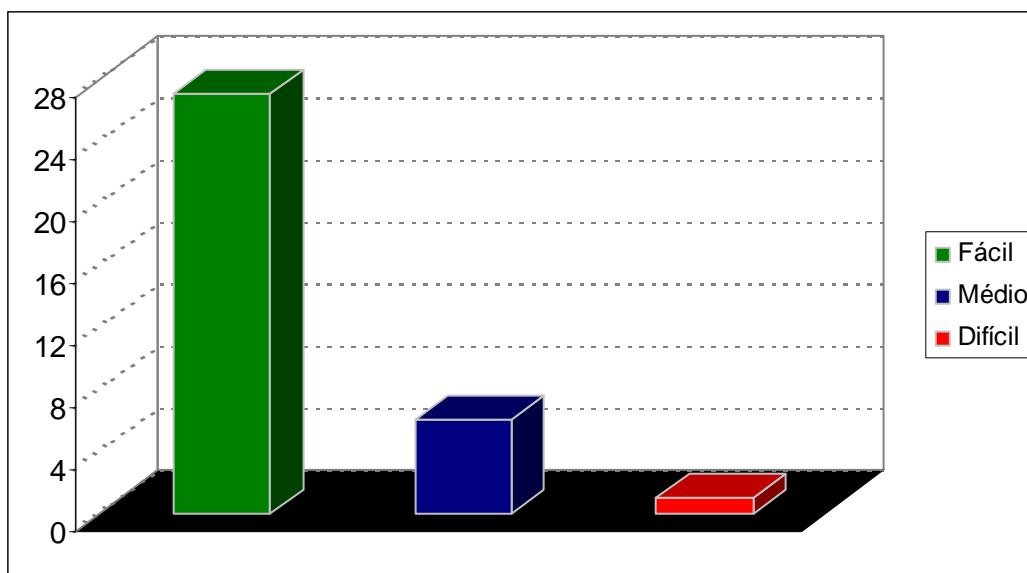


Figura 6-2: Avaliação geral do software quanto à sua utilidade.

Entretanto houve um estudante, que avaliou o jogo como difícil, conforme apresentado na tabela 6-2, devido não gostar de jogos em computador, além de não gostar de digitando via teclado pois sugeriu trocar o *chat* pela interação via voz.

3. Movimentação das peças com mouse por meio do AVD.

Conforme a Figura 6.3, uma superior quantidade de alunos apresentou facilidade quanto à movimentação das peças no AVD Quarto3D por meio do mouse. Durante as partidas foi observado que eles brincavam muito com a interação, por exemplo, alguns movimentavam as peças rapidamente enquanto olhava a ação pela tela do adversário, nesse momento o professor digitava uma mensagem via texto para que o jogador executasse jogada.

Tabela 6-3: Dados referentes à questão: movimentação das peças por meio do mouse.

		Freqüência	%	% Válido
Válido	Fácil	25	73,5	73,5
	Médio	7	20,6	20,6
	Difícil	2	5,9	5,9
	Total	34	100,0	100,0
Em branco				
Total		34	100,0	

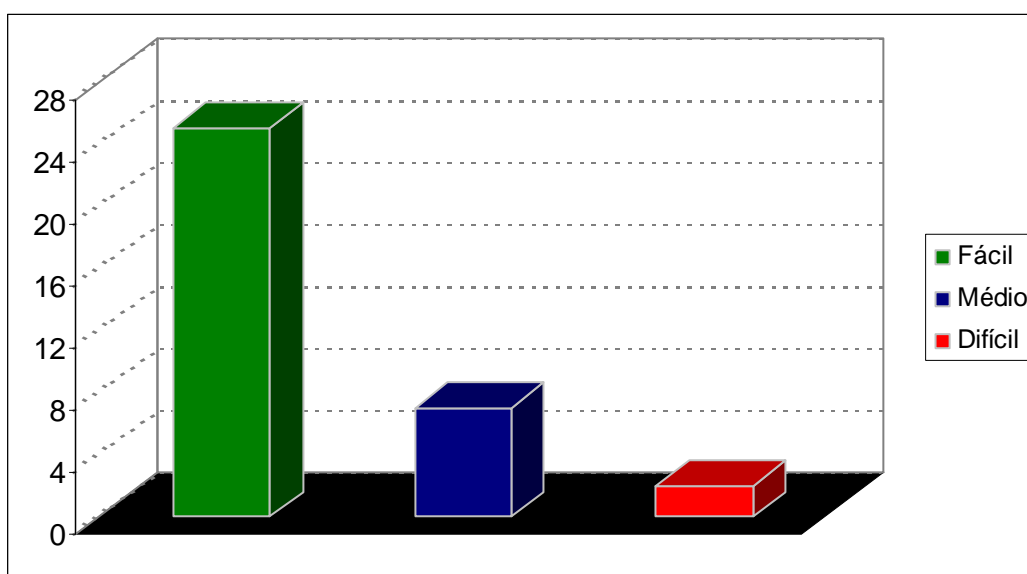


Figura 6-3: Movimentação das peças.

Embora os usuários demonstrassem grande empolgação ao verem a mobilidade das peças, houve 5,9%, ou seja, dois usuários que tiveram dificuldades com a movimentação devido falta de prática com computadores e dificuldade de entender as regras do jogo. Neste caso foi orientado novamente e apoiado nos movimento conseguindo concluir existosamente a partida. Portanto com os resultados coletados (Tabela 6-3) pode-se concluir assim que, o jogo está adequado para a faixa etária proposta.

4. Identificação de alguns ou todos os atributos das peças.

Neste item, destacou-se a identificação dos atributos (tamanho, cor, forma e furo), foram capazes de reconhecer as diferenças assim alcançando fazer o alinhamento das peças, ou seja, o objetivo do jogo (o quarto), positivamente confirma que a modelagem do jogo possui resultados satisfatórios.

Por meio de 94,1% das respostas foram positivas e demonstra que a estrutura 3D das peças, tabuleiro e opções do ambiente (câmera, movimentação e dispositivos de hardware) causou efeito positivo entre os estudantes, contribuindo como elemento fundamental na experiência da exploração do jogo.

Tabela 6-4: Dados referentes à questão: identificação de atributos.

		Frequência	%	% Válido
Válido	Sim, todos	32	94,2	94,1
	Não	1	2,9	2,9
	Alguns	1	2,9	2,9
	Total	34		
Em branco				
	Total	34	100,0	100,00

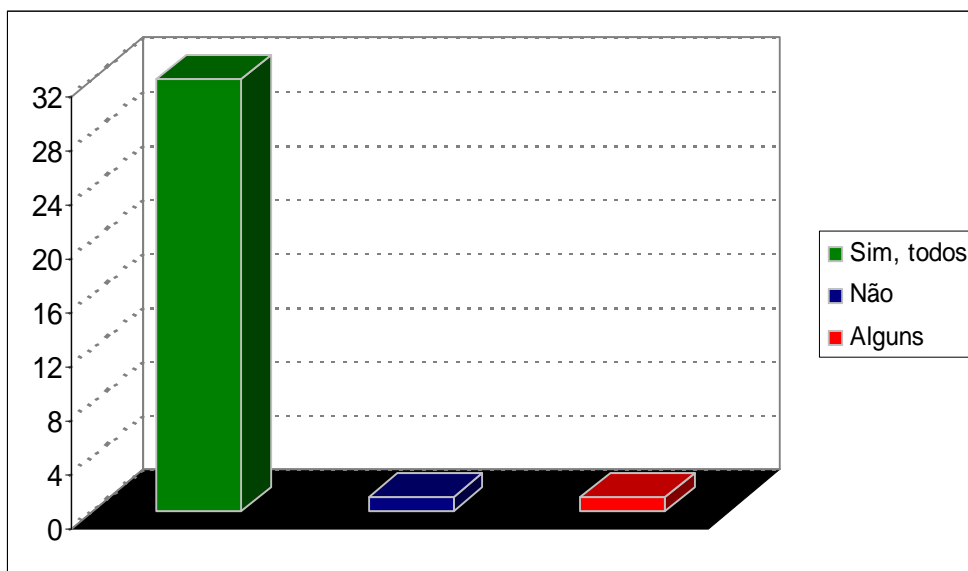


Figura 6-4: Identificação de atributos.

Em adicional (Tabela 6-4), esta questão da identificação dos atributos é importante frisar que houve 1 voto negativo devido o usuário usar óculos com grau bem avançado, além do mas o dispositivo de saída (monitor) não estava com uma imagem muito boa.

Outra resposta em “Alguns” devido ao usuário mudava a posição da câmera, onde fica difícil de identificar as diferença de atributo de algumas peças semelhante, portanto para tratar esse problema foi implementado, quando for a vez do usuário jogar a câmera se deslocará para posição de jogar.

5. Avaliação do nível de colaboração/comunicação entre os usuários.

Esta Figura 6.5 indica que o grau de percepção sobre o jogo multi-jogador online como uma ferramenta de interação entre usuários como recursos de chat, movimentação em tempo-real e representação gráfica.

Tabela 6-5: Dados referentes à questão: comunicação entre jogadores via Internet.

		Frequência	%	% Válido
Válido	Fácil	23	67,6	67,6
	Médio	10	29,5	29,5
	Difícil	1	2,9	2,9
	Total	34		
Em branco				
Total		34	100,0	100,00

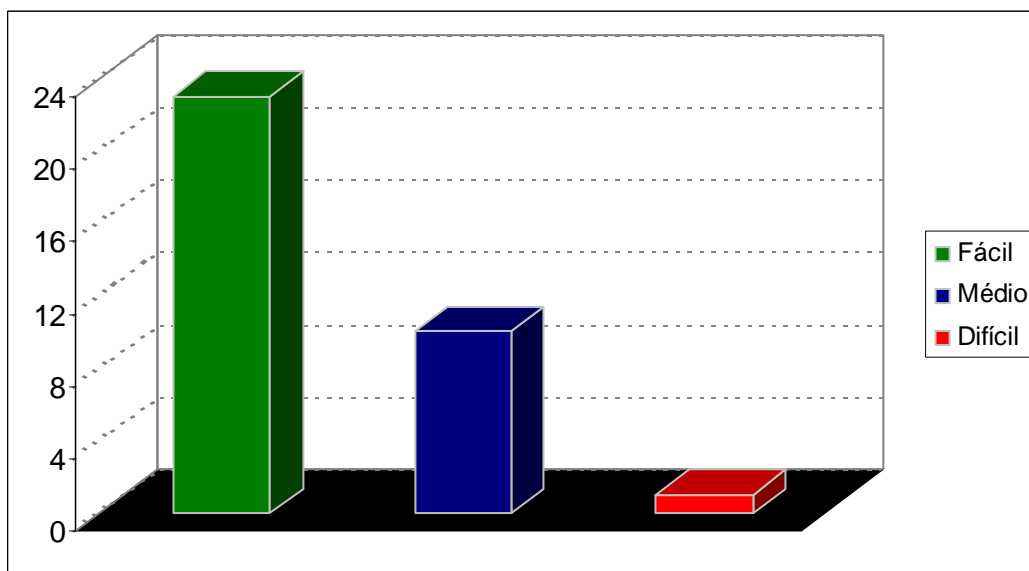


Figura 6-5: Comunicação entre jogadores.

O questionamento, buscou indagar a questões como tempo de resposta da aplicação, o chat como recurso e a representação obtendo aceitáveis resultados conforme demonstrado na tabela 6-5. Além disso, por meio desses elementos concreto comprova-se que os canais de

interação disponíveis no jogo, como *chat*, estudo de caso, repostas em tempo-real e parte gráfica, tiveram uma aplicação e uso eficiente por meio da Internet.

Embora a maioria dos estudantes teve uma percepção positiva, 29,5% tiveram uma opinião intermediária (tabela 6-5), porque alguns estudantes sugeriram o recurso de comunicação via voz.

6. Preferência sobre jogar sozinho ou não

A Figura 6-6 demonstra que a maioria das crianças tem preferência por jogar com amigos. Isto destaca a contribuição do trabalho e também a oportunidade de utilização por parte dos estudantes. Pois 5 dentre 34 participantes mencionaram que os jogos de computadores ajudam a fazer novos amigos, além de manter o contato.

Tabela 6-6: Dados referentes à questão: preferência de jogar sozinho ou não.

		Frequência	%	% Válido
Válido	Sozinho	9	26,4	28,0
	Com amigos	23	67,7	72,0
	Total	32	94,1	100,00
Em branco		2	5,9	
Total		34	100,0	

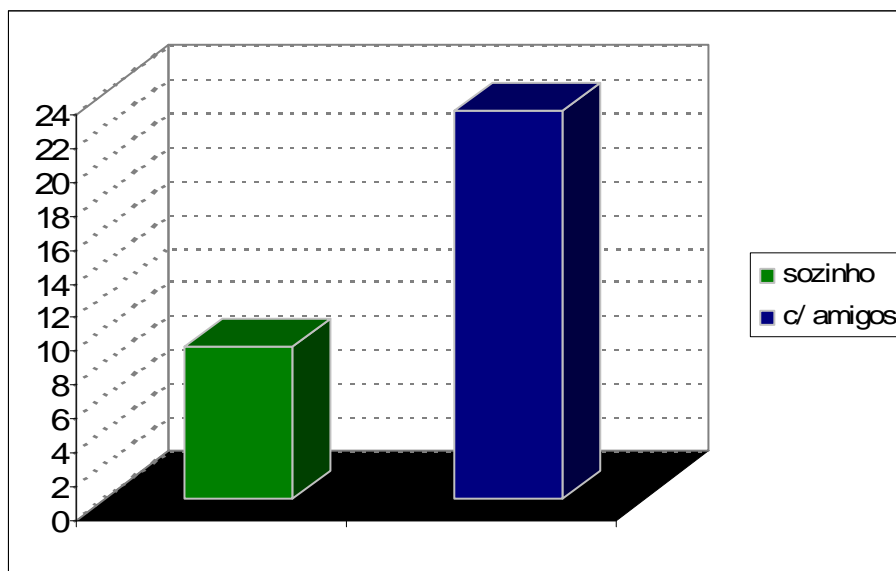


Figura 6-6: Preferência por jogar sozinho ou não.

A tabela 6-6 mostra o resultado final da compilação dos dados, no entanto possui 2 usuários onde não sabiam opinar se preferiam jogar com outros jogadores ou não. Esse ponto

e interessante influenciar o desenvolvimento de jogos tanto multi-jogador quanto jogar sozinho.

7. Aprendizado com uso AVD Educacionais

A Figura 6-7 indica que os alunos, em sua maioria, gostaram do fato de aprenderem enquanto se divertem, tendo em vista que o jogo Quarto 3D faz com que as crianças pratiquem a habilidade de observação, bem como o raciocínio lógico.

Tabela 6-7: Dados referentes à questão, aprendizado enquanto se joga no computador.

		Freqüência	%	%Válido
Válido	Muito Bom	26	76,5	76,5
	Bom	3	8,8	8,8
	Chato	5	14,7	14,7
	Total	34		
Em branco				
Total		34	100,0	100,00

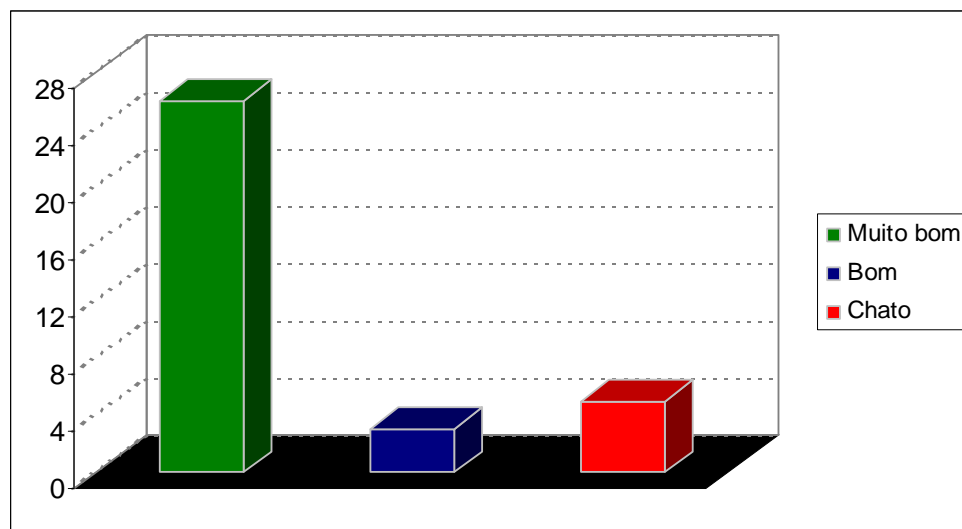


Figura 6-7: Aprender enquanto joga.

Mas conforme indicado na tabela 6-7, cinco usuários acharam chato devido a demora de adversários em efetuar suas jogadas e devido a isso, ficavam dispersos. Uma sugestão por parte dos usuários é colocar um relógio em definir um tempo para que o adversário faça sua jogada.

8. Jogabilidade com a realidade virtual

Pela Figura 6-8, pode ser observado que a maioria dos usuários, que respondeu ao questionário, considerou que o programa permitiu uma boa jogabilidade por parte dos estudantes.

A única avaliação negativa, foi devido ao estudante que tinha pouca prática com o uso do mouse, mais conseguiu completar a partida com o auxílio do professor.

Tabela 6-8: Dados referentes à questão, a jogabilidade em Realidade Virtual.

		Freqüência	%	% V Válido
Válido	Boa	29	85,3	85,3
	Regular	4	11,8	11,8
	Ruim	1	2,9	2,9
	Total	34		
Em branco				
Total		34	100,0	100,00

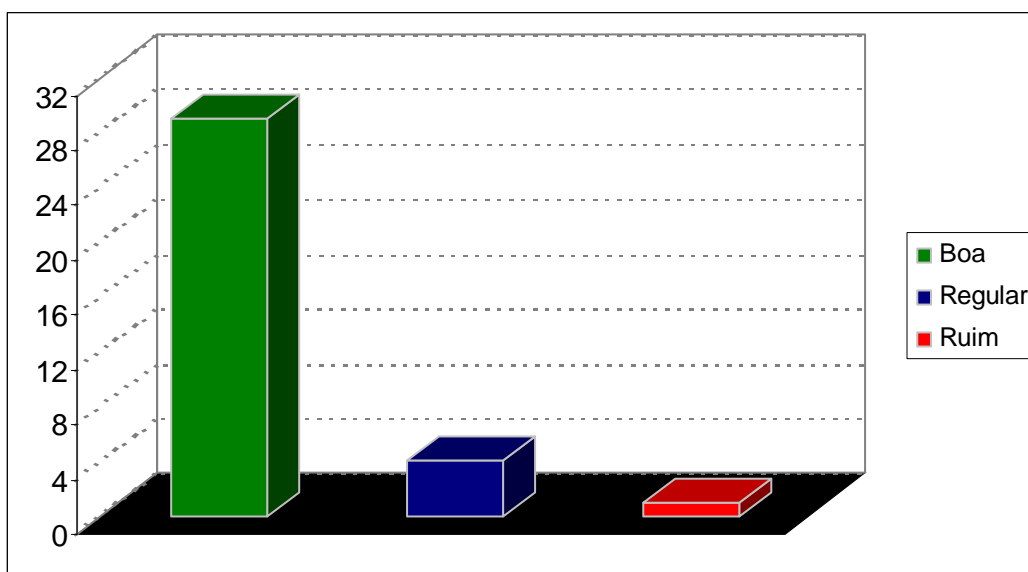


Figura 6-8: Jogabilidade em RV.

Analisando a tabela 6-8, todos os itens avaliados nos questionários, conclui-se que o software protótipo desenvolvido foi bem aceito pelos usuários entrevistados.

6.3. Considerações Finais

Este capítulo apresentou a avaliação do sistema desenvolvido para o estudo de caso. A preocupação inicial quanto ao sistema desenvolvido foi a simples eficácia do modelo, ou seja, verificar se a tecnologia escolhida propiciaria a aplicação das técnicas de Realidade Virtual para suportar jogos pela Internet.

A avaliação dos usuários parece indicar que foi alcançado o objetivo proposto, pois foi investigado o uso de técnicas computacionais que, associadas à RV, possibilitaram a criação de um jogo educacional tridimensional colaborativo com interação online. Esta verificação tem três etapas descritas a seguir:

1. Apresentando fisicamente o tabuleiro e as peças, explicando as características (cor, forma, tamanho, furo) dos objetos e as regras, foi se promovendo a introdução aos conceitos do jogo.
2. Fazendo simulações no jogo Quarto 3D por meio do computador via Internet.
3. E, por último, preenchimento de questionário para consolidação dos resultados.

De fato, o jogo Quarto 3D, criado por meio de técnicas de Realidade Virtual, mostrou-se atraente para o público alvo (aluno da 5^o Série do Ensino Fundamental) e os próprios professores consideraram-no com um grande potencial didático.

No que se refere à eficácia do software, por meio das simulações foram analisadas as características da Realidade Virtual Distribuída, por exemplo: comunicação em rede, modelo de comunicação, gerenciamento da computação e comportamento dos objetos. Quanto ao resultado, foi possível provar a aplicabilidade do jogo pelo modelo proposto.

Os professores também avaliaram positivamente o software, mostrando-se motivados quanto ao uso do sistema e sugerindo a aplicação imediata do mesmo nas escolas.

O próximo capítulo tratará das conclusões gerais do trabalho e apontará os desdobramentos possíveis do software.

6.4. Tabelas de Comparação do software proposto (Quarto3D)

Comparação dos trabalhos correlatos ao trabalho, e que fundamentam as decisões que foram tomadas em relação ao projeto.

Recursos	Sem suporte a RV sem distribuição		Com suporte a RV sem distribuição			Com suporte a RV com distribuição				
	<i>GameKid</i>	<i>Dominó Matemático</i>	<i>Planeta X</i>	<i>Supermercado</i>	<i>LudosTop</i>	<i>EducaTrans</i>	<i>Peloton</i>	<i>LVM</i>	<i>LVMRA</i>	<i>Quatro3D</i>
AVDs	<i>GameKid</i>	<i>Dominó Matemático</i>	<i>Planeta X</i>	<i>Supermercado</i>	<i>LudosTop</i>	<i>EducaTrans</i>	<i>Peloton</i>	<i>LVM</i>	<i>LVMRA</i>	<i>Quatro3D</i>
Ambiente	2D	2D	3D	3D	3D	3D distribuído Online	3D distribuído Online	3D distribuído	3D distribuído	3D distribuído Online
Simulação/Tema	Matemática	Matemática	Jogos Aventura	Educação	Jogo de Estratégia	Transito	Esporte	Biologia/Química	Biologia/Química	Jogo de Estratégia
Recursos para Interação	-	-	Efeitos	Não	Restrito	-	Dispositivos & Sensores	-	Marcadores	Chat
Modelo de Visão	Não	Não	Não	Não	Não	Assíncrona	Síncrona	Assíncrona	Assíncrona	Assíncrona
Comunicação	Local	Local	Local	Local	Local	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor	Cliente/Servidor	Cliente Servidor
Categoria	Monousuário	Monousuário	Monousuário	Monousuário	Monousuário	Multiusuário	Multiusuário	Multiusuário	Multiusuário	Multiusuário
Gerenciamento d'Computação	Não	Não	Não	Não	Não	Replicação Total	Replicação Parcial	Distribuição Parcial	Replicação Parcial	Distribuição Parcial
Técnicas RVD	Não	Não	Modelagem	Modelagem	Modelagem	Parcial	Parcial	Sim	Sim	Sim
Transmissão de Mensagens	-	-	-	-	-	Socket via Fly3D engine	Socket, IOP, Web Server	CORBA	CORBA	Socket via Torque Engine
Modelagem	EngineMaker	Flash	VRML	Direct X	VRML	Engine Fly3D	VRML	3D Max	3D Max	3D Max
Modelo de Dados	Não	Não	Não	Não	Não	Compartilhada	Compartilhada	Compartilhada	Compartilhada	Compartilhada
Autenticação	Não	Não	Não	Não	Sim	Não	Não	Não	Não	Sim
Eventos	Não	Não	Baixa	Não	Baixa	Multicasting	Multicasting	Broadcasting	Broadcasting	Multicasting

Tabela 6-9: Tabela de comparativa do software proposto.

Esta tabela 6-9 apresentou a comparação com os trabalhos correlatos, além disso a inclusão do Quatro3D software proposto desenvolvido baseado no estudo de caso LudosTop.

O resultado do software proposto e a construção da arquitetura e implementação de um ambiente virtual totalmente distribuído.

As seguintes características foram implementadas:

- Autenticação para a partida.
- Persistência automática dos dados.
- Navegação 3D em todo ambiente virtual.
- Chat para interação sobre a internet.
- Interface 3D para uso de teclado ou mouse.
- Aplicação de técnicas de Realidade Virtual Distribuída.
- Jogar via Internet, Intranet ou sozinho.

Durante as pesquisas, foi descoberto que o desenvolvimento gráfico com uma engine enriquece a experiência do usuário, pois permite a criação de complexos ambientes virtuais.

O desenvolvimento do Quatro3D foi modular dividido em: interface (GUI), ambiente virtual do jogo e distribuidor de eventos. A aplicação possui a integração de interface e o núcleo que controla as regras do jogos mais envios/recebimentos das alterações efetuadas.

Portanto foi examinado a sua eficiência juntamente com usuários dentre 11 e 12 anos e obteve resultados aceitáveis. Os benefícios do resultado e a construção de um ambiente virtual distribuído para internet que seja multi-jogador combinado com métodos e técnicas para melhorar a performance da latência sem dependência da lagura de banda provida pela solução.

Capítulo 7

7. Conclusões e Trabalhos Futuros

7.1. Introdução

Este trabalho partiu da consideração de que o avanço tecnológico ligado às técnicas de simulação e animação, suportadas em Realidade Virtual Distribuída, representa hoje, para o campo da educação, a possibilidade extraordinária de veicular os conteúdos educativos de forma acessível, ágil e atraente.²³

Considerou-se também o grande interesse das crianças pelos jogos de computador e videogames.

Com base neste contexto cultural e tecnológico, a idéia foi produzir um *software* educacional que, estruturado a partir das técnicas de Realidade Virtual e Técnicas de Distribuição, pudesse disponibilizar um jogo interativo multi-jogador pela Internet.

Identificou-se no jogo “Quarto” um interessante estudo de caso para a implementação de tais técnicas para gerar o *software*.

Em seguida, foram pesquisadas determinadas técnicas de Realidade Virtual e Realidade Virtual Distribuída, optando-se pelo motor de jogos Torque e API OpenGL como meio de implementação do sistema.

O passo seguinte foi analisar *softwares* educacionais. O objetivo foi verificar o material já disponível, observar os aspectos pedagógicos destes programas, o design, os aspectos de implementação (recursos computacionais utilizados), bem como aproveitar os resultados positivos dessas experiências computacionais.

²³ Somado a essas técnicas de RV, está a possibilidade de disponibilizar tais conteúdos via Internet. Reconhecendo estas possibilidades, como já foi dito, o Ministério da Educação e Cultura tem investido no sentido de dotar as escolas de uma infra-estrutura que permita disponibilizar para as escolas o material produzido com base em tal tecnologia.

Feito isso, partiu-se para a elaboração do *software*, buscando-se os objetivos previamente definidos.

7.2. Conclusões

Acredita-se que o objetivo tenha sido alcançado, ou seja, a construção um *software* educacional que integrasse técnicas de Realidade Virtual, distribuição e estratégias de jogos, um *software* que fosse lúdico e pedagogicamente fundamentado à aplicabilidade do estudo de caso “*LudosTop*” (FREITAS, 2008).

A reação das crianças que utilizaram o programa parece confirmar que tal sistema protótipo está no caminho certo. Isso foi demonstrado pelo modo como responderam ao desafio de distinguir os atributos das peças do jogo ou de aplicar as estratégias solicitadas: se tivesse sido extremamente fácil identificar os atributos das peças ou utilizar as estratégias propostas (alinhar peças), significaria que o jogo estaria inadequado à faixa etária prevista; se tivesse sido muito difícil, estaria, de modo análogo, inadequado para tal faixa etária.

Outro aspecto que indica que o sistema protótipo está no caminho certo é o fato de as crianças envolvidas terem tido relativa facilidade para manuseá-lo. Quanto a este aspecto, aliás, as dificuldades de manuseio apontadas pelas crianças serão úteis para o aperfeiçoamento do programa.²⁴

O entusiasmo demonstrado pelas crianças em lidar com um programa em Realidade Virtual e que possibilita a interação, também indica que o *software* alcançou o objetivo proposto, qual seja, o de envolver as crianças justamente por seu aspecto lúdico. Isto demonstra também a adequabilidade das técnicas de Realidade Virtual, tais como imersão e navegação, no sentido de que propiciam a construção de ambientes interessantes de aprendizagem.

Em relação aos aspectos tecnológicos, para os AVDs é fundamental a comunicação em rede, por esse motivos esse modelo apresenta uma satisfatória melhoria, em sincronização com usuário de diferentes largura de banda, problemas de empacotamento de entidades, confiabilidade em sistema distribuídos e falta de autenticação em aplicações distribuídas de Realidade Virtual. As principais vantagens são os controles de envio e recebimento de dados;

²⁴ A análise das crianças a respeito da facilidade de manuseio está comentada no cap. 6.

com um framework de desenvolvimento e menos integrações entre diferentes plataformas demonstrar um potencial de significado para área de AVDs.

As desvantagens que o *socket* demonstrou menos escalavel perante as outras tecnologias de distribuição (*CORBA*, *RMI* e *RPC*), pois o *socket* exige um maior conhecimento do mecanismo de comunicação em uma rede de computadores, embora todas as etapas necessárias para criação de uma aplicação cliente-servidor, devem ser criadas pelo programador, desde realizar as conexões até o controle de fluxo de envio/recebimento. Tal fato acarreta em mais trabalho de programação.

Finalmente, um outro objetivo foi alcançado de disponibilizar, um *software* de código aberto, que funcione na internet sem dependência da largura de banda do usuário e demonstra que empregando as técnicas de ambientes virtuais distribuídos pode-se melhorar à performance e conseqüentemente desvendando métodos para o avanço da escalabilidade em sistemas de RVD.

7.3. Trabalhos Futuros

Estudos futuros poderão tornar o sistema ainda mais eficiente. A seguir, são relacionadas sugestões neste sentido:

- Implementação de algoritmos relacionados à (IA) Inteligência Artificial na lógica do software proposto, embora o jogo tem a finalidade ser estratégico, ou seja, praticas de observação e concentração, pode-se atribuir níveis de dificuldades quando o usuário queira jogar sozinho.
- Pesquisa em balanço de carga (*Load balancing*) para melhorar a escalabilidade, o serviço de processamento de uma aplicação consome excessivamente de uma aplicação, prejudicando em distribuição da informações, pois por meio de pesquisas para compartilhar a carga de processamento, melhore o tempo de resposta entre usuário e sistema.
- Criar juntamente com a Torque Engine bibliotecas para Realidade Virtual e Realidade Virtual Distribuído para acelerar e facilitar o desenvolvimento de

aplicações distribuídas de RV já agregando as técnica e métodos de distribuição junto aos Ambientes Virtuais.

- Desenvolver mecanismo de transmissão de pacotes de atualizações de jogadas das partidas encriptada via Internet, para evitar ataques hackers durante uma partida.
- Adaptação do jogo para dispositivos móveis, com a tendência da mobilidade constante e o aperfeiçoamento dos celulares/terminais seria uma oportunidade para oferecer a possibilidade de jogos multi-jogadores;
- Disponibilizar o jogo de forma distribuída com o uso de VOIP na comunicação, tanto para computadores convencionais quanto à celulares.
- Desenvolver a possibilidade de criar avatares, em conjunto com uma variedade de jogos tabuleiro, como: dama, quebra-cabeças, xadrez e outros. Assim o usuário poderia caminhar em um parque e participar de vários tipos de jogos ao mesmo tempo com diferentes adversários remotamente.
- Criar mecanismos de modo que os professores possam analisar o nível do desenvolvimento das jogadas efetuadas por meio de geração de arquivos como históricos da partida, assim utilizam-las como métricas de avaliação;

7.4. Considerações Finais

Pesquisando os existentes Ambientes Virtuais Distribuídos para aprender sobre as implicações do design, obstáculos técnicos, assim como os *frameworks* para o desenvolvimento da uma aplicação 3D multi-usuário para a Internet.

Com isto, foi possível obter uma visão geral de como construir e integrar em diferentes tipos de redes, examinando: desempenho, representação gráfica, mecanismos de eventos, compartilhamento, controle de acesso e interfaces gráficas.

Foram feitas análises em diversos ambientes de aprendizagem, e revelaram um conjunto de pontos interessantes. Os pontos avaliados são os forma

Revelou que os atuais sistemas distribuídos não usam técnicas de melhoria em comunicação & transmissão para Internet, além de usarem um modelos distribuídos sem nenhuma ferramenta de interação tão pouco autenticação.

O resultado desta dissertação e a criação de uma arquitetura cliente-servidor baseado em trabalhos correlatos usando os princípios de design para produzir um software para multiusuário interativo em tempo-real para a Internet desenvolvendo esses mecanismos.

Enquanto protótipo, portanto, considera-se que este trabalho cumpre seu propósito como experiência de um modelo de distribuição com suporte a Internet, por meio técnicas de Realidade Virtual Distribuída melhorar a comunicação em rede e conseqüentemente na interação homem-máquina, além de representar uma possibilidade muito interessante de enriquecimento e amadurecimento profissional.

Referências Bibliográficas

ADAMS, L. **Visualização e realidade virtual**, Ed. Makron Books, pp. 255-259, São Paulo, 1994.

ABED, A. L. Z. **O jogo de Regras na Psicologia Clínica: Explorando suas possibilidades de Uso**. In:___ Org. MASINI, Elcie. Ação Psicopedagogia: II ciclo de estudos de psicopedagogia Mackenzie. São Paulo: Memnon. 2000.

ASSIS, G.; FICHEMAN, I.; CORRÊA, A.; NETTO, M.; LOPES, R. **EducaTrans: um jogo educativo para o aprendizado do trânsito**. RENOTE. Revista Novas Tecnologias na Educação, v. 4, p. 1-10, 2006.

AKENINE-MOLLER, T.; HAINES, E. **Real Time Rendering**. A K Peters, Massachusetts, 1999.

ARAUJO, R.B. e KIRNER, C. - **Arquitetura de um Ambiente Virtual Distribuído para Aplicações de Larga Escala**. Anais do XIII Simpósio Brasileiro de Redes de Computadores, Belo Horizonte, MG, Maio de 1995, pp. 431, 441.

ARAUJO, R. B. e KIRNER, C. **Especificação e Análise de um Sistema Distribuído de Realidade Virtual**. Anais XIV Simpósio Brasileiro de Redes de Computadores – SBRC, SBC, Fortaleza-CE, Maio, 1996.

ARAUJO, F. V.; CORDENONSI, A. **Planeta X – Desenvolvimento de um Jogo Virtual Utilizando a Tecnologia VRML**. 8 th Taller Internacional de Software Educativo. Santiago, Chile. Universidade Luterana do Brasil 2003. Disponível em: http://www.tise.cl/archivos/tise2003/papers/planeta_x.pdf. Acesso 19 Janeiro 2008.

ARAUJO, R. B.; BATTAIOLA, A. L.; GOYOG, C. G. **Exploração do uso de Realidade Virtual no Aprendizado de Habilidades Acadêmicas para o Ensino Fundamental**. Disponível em: <<http://www.inf.ufsc.br/sbc-ie/revista/nr4/sbie01-araujo.htm>>. Acesso 09 Novembro 2005.

ATTROT, W.; TRAMONTINA, G. **Sockets, RPC, RMI e CORBA: Uma comparação dessas tecnologias para comunicação cliente-servidor**. Universidade Estadual de Campinas – UNICAMP, Outubro, 2002.

BATTAIOLA, L. **Jogos por computador - Histórico, Relevância Tecnológica e Mercadológica, Tendências e Técnicas de Implementação**. In: Jornada de Atualização em Informática, 19, 2000, Curitiba. Anais... Curitiba: PUCPR, 2000.

BARNETT, M.; SHIH, P. **VR – History and and Technologies**. URL: <http://inkido.indiana.edu/a100/definitions.html>. 1998. Acessado em 22 de abril de 2008.

BEGAULT, D. R. **3-D Sound for virtual reality and multimedia**, Academic Press, Cambridge, MA, 1994.

BINDA, J.; SILVA, K.; PASSOS, O. **Implementação De Jogos Educacionais por meio de Engines De Computação Gráfica**. Escola Superior de Tecnologia do Amazonas, 2005.

BOTELHO, F. **Realidade Virtual Aplicada a Aquisição de Informações do Subsolo**. Grupo de Realidade Virtual e Multimídia do Centro de Informática – UFPE, Junho de 2002. Disponível em: <<http://www.cin.ufpe.br/~grvm>>. Acesso em 5 Setembro 2007.

BOWMAN, D. A., *et al.*. **3D User Interfaces: Theory and Practice**. Boston: MA: Addison-Wesley. 2005.

BRAS, J. **Realidade Aumentada: Uma visão e duas metodologias de realidade melhorada**. <http://planeta.clix.pt/jbraz_home/docs/Projectos.htm>. Acesso em: 02 Jan 2008.

BRILL, L. **Virtual Auditoriums – Sharing VR in Small Groups**. Virtual Reality Special Report, Nov. 1994, pp.17.

BROOKS, F. **Project GROPE - Haptic Displays for Scientific Visualization**. ACM Computer Graphics, 24(4):177-185, Aug. 1990.

BROOKS, F. **Walkthrough Project: Final Technical Report to National Science Foundation Computer**, Information Science and Engineering. 1992.

BURDEA, G. e COIFFET, P. **Virtual Reality Technology**, 2nd Edition, Wiley, 2003

BYRNE, C. M. **Water on Tap: The Use of Virtual Reality as an Educational Tool**, Department of Industrial Engineering, University of Washington, 1996.

BYRNE, C. M. **The Use of Virtual Reality as Educational Tool**. Washington University. Disponível em <<http://www.hitl.washington.edu/publications>>. Acesso em: 22 de abr. 2008.

CAMACHO, M. **Realidade Virtual e Educação**. Lisboa, Universidade Aberta. Disponível em <<http://phoenix.sce.fct.unl.pt/simposio/30.html>>. Acesso em: 23 Ago. 2006.

CAMELO, M. **Ambiente em Realidade Virtual para Usuários de Educação a Distância Estudo da Viabilidade Técnica**. Dissertação de Mestrado. Universidade Federal de Santa Catarina. Santa Catarina-SC, 2001.

CARR, K; ENGLAND, R. **Simulated and Virtual Reality**. London: Taylor and Francis. 1995.

CARDOSO, A.; KELNER, J.; KIRNER, C.; LAMOUNIER, E. **Tecnologias para o desenvolvimento de sistemas de realidade virtual e aumentada**; Editora Universidade da UFPE. Recife – PE. p.4. 2007.

CARDOSO, A.; LAMOUNIER, E. **Realidade Virtual Uma abordagem Prática**, Ed. Mania do Livro, 2004.

CARDOSO, A.; LAMOUNIER, E. **A Realidade Virtual na Educação e Treinamento.** Pré-Simpósio : VII Symposium on Virtual Reality, Editora Mania de Livro. São Paulo – SP. p. 256/264. 2004.

CARLSON, C e HAGSAND, O. **DIVE – A Multi-User Virtual Reality System.** Proc. of The IEEE VRAIS'93, IEEE, 1993, pp. 394-400.

CATER, J. **Use of the Remote Access Virtual Environment (RAVEN) for coordinated IVA-EVA Astronaut Training end Evaluation.** Presence, v.4, n.1, p.103-109, 1995.

CODDELA, C et al.,. **A Toolkit for Developing Multi-User Distributed Virtual Environments.** In: IEEE Virtual Reality Annual International. 1993, Washington, DC, USA. Proceedings of the IEEE VRAIS'1993. p.401-407.

COSTA, R. M.; et al.,. **Ambientes Virtuais na Reabilitação Cognitiva.** I Workshop de Realidade Virtual e Visualização Científica do Laboratório de Métodos Computacionais em Engenharia – UFRJ – Rio de Janeiro – Brasil, 2002.

CORREIA, R. C. M. **AVDNet – Arquitetura para Ambientes Virtuais Distribuídos Escaláveis baseada na Infra-Estrutura Atual da Internet.** 2005. 131f. Tese de Doutorado – Instituto Tecnológico de Aeronáutica, São Jose dos Campos.

COSTIKYAN, G. **Why online games suck (and how to design ones that don't).** Disponível em: <<http://www.costik.com/onlinsux.htm>>. Acesso em: 01 Nov 2003.

COULOURIS, G. DOLLIMORE, J.; KINDBERG, T., **Distributed Systems: Concepts and Design.** 4th Edition. Addison-Wesley/Person Education, 2005, p.994.

CRUZ-NEIRA, C. et al.,. **The CAVE Audio Visual Experience Automatic Virtual Environment.** Communication of the ACM, 35, pp. 64-72., 1992.

CYBEREDGE . **The Virtual Lexicon a Glossary Of VR Terms.** URL: <<http://www.cyberedge.com/4a1.html>>. Acessado em 22 de maio de 2006.

DEDE, C. **The Evolution of Constructivist Learning Environments: Immersion in Distributed, Virtual Worlds.** Educational Technology 35(5), 46-52, 1995.

DIAS, W. **Ambiente de Desenvolvimento de Manufatura Virtual.** Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal de Itajubá – Minas Gerais, 2006.

FREITAS, M. R. et al.,. **Proposta de Integração de Jogos e Realidade Virtual no Ensino Aprendizagem de Matemática** - Projeto Ludostop. Conferência Internacional em Educação em Engenharia e Computação, Santos - SP. 2007. CD-ROM.

FREITAS, M. R. **LudosTop: Estratégia de Jogos e Realidade Virtual com vistas ao Desenvolvimento do Pensamento Lógico-Matemático.** Dissertação de mestrado. Universidade Federal de Uberlândia, 2008.

FURNESS, T. **The super cockpit and its human factors challenges**. Proceedings of the Human Factors Society, 30, 1968, pp. 48-52.

GATES, B. **A Estrada do Futuro**. Companhia das Letras. São Paulo, 1995.

GIANPAOLO, U., MAURICIO, C., JOHN, T. e ROBERT, E. **The Peloton Bicycling Simulator**. In Proceedings of 3rd Symposium on the Virtual Reality Language VRML 98, ACM SIGGRAPH, 1998.

GONÇALVES, J. P. **Uso de Jogos Computacionais Educativos Via Internet na Educação – Projeto FORMEL**. Projeto de iniciação científica, Universidade Estadual de Campinas (UNICAMP). Campinas. 2001.

GRADECKI, J. **The virtual reality construction kit**, John Wiley & Sons, 340 Pp., 1995.

GUSMÃO, L. **Escolas públicas: governo quer universalizar acesso à Internet até 2010**. G1, em Brasília. Disponível em: <http://g1.globo.com/Noticias/Vestibular/0,,MUL13610-5604,00.html>. Acesso: 04 mar. 2007.

HAND, C. **A Survey of 3D Interaction Techniques**; <http://www.cms.dmu.ac.uk/~cph/Publications/CGF97/3dint.ps.gz>. Acesso: 08 Mai 2007.

HARRISON, D.; JAQUES, M. **Experiments in Virtual Reality**. Digital Press. 1996.

HEBERT, A; CHEN, A. **A new collaborative software package: TeamSpace at Stanford University**. In: User Services Conference, 33, 2005, Monterey, CA, USA. Proceeding of the 33rd annual ACM SIGUCCS conference on User services. New york, USA: ACM Press 2005, p109-112.

HEETER, C. **Implications of interactivity for communication research**, in Media Use in the Information Age: Emerging Patterns of Adoption and Consumer Use, Jerry Salvaggio and Jennings Bryant, eds., Lawrence Erlbaum Associates, 217-235. 1989.

HEIN, M. **Metaphysics of Virtual Reality**. Oxford Univ. Press. 1994.

HILTZ, S. e WELLMAN, B. **Asynchronous Learning Networks as a Virtual Classroom**, Communications of the ACM. vol 40,9. Setembro, 1997.

IDS SOFTWARE, **Quake III: Arena**, <http://www.idsoftware.com>. Acessado em 10 de julho de 2008.

JACOBSON, L. **Realidade virtual em casa**. Rio de Janeiro, Berkeley, 1994.

JONES, S. **Let the games begin: Gaming technology and entertainment among college students**. Disponível em http://www.pewInternet.org/reports/pdfs/PIP_College_Gaming_Reporta.pdf. Acesso em 05 mai 2008.

JUNIOR, O. **LRVCHAT3D: Desenvolvimento de um Ambiente Virtual Tridimensional Multi-usuário para Internet**. Dissertação (Mestrado em Engenharia de Produção), Universidade Federal de Santa, 2003. 107 p.

KATO, Y. **Introduction: What is force feedback devices?** <http://ldt.stanford.edu/~yasukato/portfolio/class/cs147/as8/>. Acessado em 21 de junho de 2008.

KIRNER, C., IPÓLITO, J. "Projeto de Ambientes Virtuais Multi-Usuários Usando Java e VRML". Proceedings Workshop on Virtual Reality (2000), Gramado, p.169-179.

KIRNER, C.; TORI, R. **Realidade Virtual: Conceitos e Tendências**. Pré-Simpósio: VII Symposium on *Virtual Reality*. São Paulo: Mania de Livro. 2004. p.354.

KIRNER, C. **Virtual Environments for Shared Interactive Visualization**. Paper Accepted for Presentation at the Workshop of the German-Brazilian Cooperative Programme in Informatics, Berlim, 14-15 December 1995.

KIRNER, C. **Apostila do ciclo de palestras de realidade virtual**, Atividade do Projeto AVVIC- CNPq (Protem - CC - fase III) - DC/UFSCar, São Carlos, pp. 1-10, Out., 1996.

KIRNER, C. **Realidade Virtual: Dispositivos e Aplicações**. VII Escola de Informática da SBC Regional Sul, 10 a 14 Maio/1998, pp. 135-158.

KIRNER, C. **Sistemas de Realidade Virtual**. Disponível em: <http://www.dc.ufscar.br/~grv/tutrv/tutrv.htm>. Acesso 24 Abril. 2008. Acesso em 06 de Janeiro de 2008.

KOZOVITS, L. E. **Arquitetura para Jogos Massive Multiplayer**. Universidade Católica do Rio de Janeiro. 2003.

KIRNER, C.; SISCOOTTO, R. A. **Fundamentos de Realidade Virtual e Aumentada**. Projetos e Aplicações. Porto Alegre: SBC. 2007.

KRUEGER, M. **Small Planet**. <
http://www.iamas.ac.jp/interaction/i97/artist_Krueger.html. > Acesso em 10 de Maio de 2008.

KUROSE, J. F.; ROSS, K. W. **Redes de Computadores e a Internet: Uma nova abordagem**, 1.ed. São Paulo: Addison Wesley, 2003. 548p.

LIMA, L. O. **Piaget: Sugestões aos Educadores**. Petrópolis: Vozes. 2000. 253 p.

LUZ, E. **Educação a Distância e Educação Matemática: contribuições mútuas no contexto teórico-metodológico**. Tese (Doutorado em Engenharia de Produção), Universidade Federal de Santa Catarina - Florianópolis, 2003.

MACHADO, L. **Conceitos Básicos da Realidade Virtual**, São Paulo, 1995 <<http://www.lsi.usp.br/~liliane/conceitosrv.html>> Acesso 01 mai 2008.

MACEDO, L. **Aprender com Jogos e Situações-Problema**. Porto Alegre: Artmed. 2000. 116 p.

MACEDO, L. **Avaliação de Arquitetura para Jogos de Ação Distribuídos Desenvolvidos em Corba**. Universidade Federal de Pernambuco. Recife, PE, Brasil, 2005.

MACEDONIA, M. **NPSNET: A Network Software Architecture for Large-Scale Virtual Environments**. Presence, Vol 3, No. 4, Fall 1994, pp. 265-287.

MACEDONIA, M. **A taxonomy for Networked Virtual Environments**. <<http://doi.ieeecomputersociety.org/10.1109/93.580395>> Acesso em 16 Mai 2008.

MANNINEN, T. **Interaction forms and communicative actions in multiplayer games**. The International Journal of Computer Games Research., artigo 031. Disponível em: <<http://www.gamestudies.org/0301/manninen>. Acesso 23 mai 2008.

MANSSOUR, I. **Introdução ao OpenGL**. <<http://www.inf.pucrs.br/~manssour/OpenGL/Introducao.html>> . Acesso 13 Jul 2008.

MIZUKAMI, M. **Ensino: as Abordagens do Processo**. São Paulo: EPU. 1986. 119p.

MOREIRA, M. **Teorias de Aprendizagem**. São Paulo: EPU. 1999. 195 p.

MORTON, D. **Study sounds alarm over video game use: Isolation, helplessness characterize the world of heavy player, study finds**. Disponível em: <<http://www.sfu.ca/mediapr/sfnews/1998/April2/kline.html>>. Acesso 03 fev 2008.

NASA; **Virtual Wind Tunnel – Space Shuttle CFD**. <<http://www.nas.nasa.gov/Software/VWT/>>. Acessado em 14 de abril de 2007.

NUNES, C. **Uma Aplicação de Técnicas de Realidade Virtual Não-Imersiva no Processo Multidisciplinar de Educação**. 2002. 72f. Dissertação (Mestrado em Ciências – Realidade Virtual) Depto. Laboratório de Computação Gráfica, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, UFU, Uberlândia, 2002.

PANTELIDIS, V. S. **Reasons to use Virtual Reality in Education**. Disponível em: <<http://www.thejournal.com/articles/14425>>. Acesso 09 maio. 2007.

PARALLELGRAPHICS. **Cortona VRML Plug-in Company**.; Disponível em: <http://www.parallelgraphics.com/products/cortona/notes/>. Acesso 28 dez. 2007.

PAUSCH, R., SNODDY, J., TAYLOR, R., WATSON, S., HASELTINE, E.; **Disney's Aladdin: first steps toward storytelling in virtual reality**. In Proceedings of SIGGRAPH'96. 1996. ACM.

PINHO, M.; **Realidade Virtual como Ferramenta de Informática na Educação**. In: 10th Simpósio Brasileiro de Informática na Educação, 1996, Belo Horizonte, MG. Anaisth SBC - Simpósio Brasileiro de Informática na Educação: Belo Horizonte, SBC, 1996.

POSITIVO, I. **Introdução sobre o software Supermercado 3D**; Disponível em: http://www.positivoinformatica.com.br/site/ed_soft_interna.asp?id_software=59. Acesso em mai.2008.

Parâmetros Curriculares Nacionais: Matemática. Disponível em: <http://mecsrv04.mec.gov.br/sef/estrut2/pcn/pcn1a4.asp>. Acesso jan. 2007.

HAWKES, R. A. **Software Architecture for Modeling and Distributing Virtual Environments**. 1996. 187f. Tese (Doutorado em Ciência da Computação) – Univesity of Edinburg, Edinburg, 1996.

RIBEIRO, W. **Arquitetura para Distribuição de Ambientes Virtual Multidisciplinares**. 2006. 159f. Tese (Doutorado em Ciências – Realidade Virtual) Depto. Laboratório de Computação Gráfica, Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, UFU, Uberlândia, 2006.

RIZZI, C. B.; *et al.*. **Experimentos Computacionais com Grupamentos Lógicos Matemáticos de Jean Piaget**. XIV Simpósio Brasileiro de Informática na Educação, Rio de Janeiro. Disponível em: <http://www.nce.ufrj.br/sbie2003/publicacoes/paper51.pdf>. Acesso 06 jun 2007.

ROBERTSON, G. G. et al. **Non-immersive virtual reality**, IEEE Computer, pp. 81-83, Feb., 1993.

ROERL, B. **Some Thoughts on Behaviour in VR Systems**. Disponível em: <http://sunee.uwaterloo.ca> > Acesso em 29 Maio 2006.

SCHREIER, M. **Audio Server for Virtual Reality Applications**. Universidade de Stuttgart, Alemanha. Dissertação de mestrado em Engenharia Eletrônica e Computação. Disponível em: <http://www.hlrs.de/organization/vis/velab/AudioServer/files/AudioServer.PDF>. Acesso 17 Maio 2008.

SEMENTILLE, A. C. **A utilização da Arquitetura CORBA na Construção de Ambientes Virtuais Distribuídos**. 1999. 186f. Tese (Doutorado em Física Computacional) – Instituto de Física de São Carlos, 1999.

SILVA, W.; WAGNER, M. **Ambientes Virtuais Distribuídos Multidisciplinares Usando Realidade Aumentada**. Universidade Federal de Uberlândia – MG. Dissertação de mestrado em Engenharia Elétrica, 2008.

SIMGRAPHICS. **Vactor Performer - An Introduction**. <http://www.simg.com/pdf/PerformerIntro.pdf> >. Acessado em 09 de janeiro de 2006.

SINGHAL, K. S.; CHERITON, D. **Using a position history-based protocol for distributed object visualization**. Technical Report Department of Computer Science, Stanford University, 1994. 26 p.

SINGHAL, S.; ZYDAL, M. **Networked Virtual Environments: Design & Implementation**. Addison Wesley, Boston, 1999.

SMED, J.; KAUKORANTA, T.; HAKONEN, H. **Aspects of networking in multiplayer computer games**. In: Proceedings Of International Conference An Application And Development Of Computer Games In The 21st Century, 74-81, 2001.

SANCHO, J. **Para uma Tecnologia Educacional**. Tradução Beatriz A. Neves. Porto Alegre: ArtMed. 1998. 327 p.

SILVA, J. C. **Aprendizagem Mediada por Computador: uma proposta para desenho técnico mecânico**. Tese (Doutorado em Engenharia de Produção), Universidade Federal de Santa Catarina, 2001. 206 p.

SILVA, Â. M.; *et al.*. **Guia para normalização de Trabalhos Técnicos-Científicos: projeto de pesquisa**, trabalhos acadêmicos, dissertações e teses. 5. ed. Uberlândia: EDUFU. 2005.

SILVA, L. F. **Associando Realidade Virtual não-imersiva e Ferramentas Cognitivas para o Ensino de Física**. Dissertação (mestrado em Ciências), Universidade Federal de Uberlândia, 2006. 116p.

STYTZ, M. R. **Distributed Virtual Environment**. IEEE Computer Graphics and Applications, Los Alamitor, CA USA, v.16, n.3, p.19 – 31, may 1996.

SUTHERLAND, I. **A head-mounted three dimensional display**. Proceedings of the Fall Joint Computer Conference, 1968.

SUN, MICROSYSTEMS. **The Java Material**. <<http://java.sun.com/docs/books/tutorial/networking/sockets/definition.html>> Acesso em 14 Jul 2008.

SCHUMANN, David W.; THORSON, Esther. **Advertising and the world wide Web. Mahwah**: Lawrence Erlbaum, 1999.

SZWARCMAN, D.; FEIJÓ, B; COSTA, M.. **Goal-oriented dead reckoning for autonomous characters**. Computer & Graphics, 5(6):999-1011, 2001.

TEWS, R. **Archetypes on acid: Video game and culture**. In J Wolf (Ed.), The medium of the video game (pp. 169-182). Austin: University of Texas Press.

TURUNEM, M. **Error Handling in Speech User Interfaces in the Context of Virtual Worlds**, <http://www.cs.uta.fi/~mturunen/error_handling/error_handling.html>. Acesso em 07 Jul 2008.

TORQUE, E. **The Engine to development of games**. <<http://www.garagegames.com/>>. Acesso em 07 Jul 2007.

VASCONCELOS, R. S.; *et al.*. **Dominó Matemático: Uma proposta de Jogo Educativo Eletrônico para o Ensino de Matemática**. Anais do WJOGOS 2005 – IV Workshop Brasileiro de Jogos e Entretenimento Digital. São Paulo. p. 280/283. 2005.

WLOKA, M. **Lag in Multiprocessor VR**. Presence, v.4 n.° 1, Spring 1995.

WRIGHT, R.; SWEET, Michael. **OpenGL SuperBible**. 2nd ed. Indianapolis, Indiana: Waite Group Press, 2000. 696 p.

ANEXOS

ANEXO A – Questionário

Sua Idade: _____ Série: _____ Sexo: () M. masculino. () F. feminino.

Escola: _____

Atenção: Por favor, não deixe nenhuma questão sem resposta.

Obrigado

Informações Gerais

1. Tem computador em sua casa? () Sim () Não
2. Tem Internet em sua casa? () Sim () Não
3. Você costuma usar o computador para fazer o que?
() Para jogar () Para fazer pesquisa na Internet () Para brincar
() Outros, quais? _____
4. Você joga no computador? () Sim () Não
5. Quanto tempo você costuma jogar ? () 1-2 hs () 2-4 hs
6. Você joga pela Internet com seus amigos ao mesmo tempo? () Sim () Não
7. Que tipo de jogos na Internet você costuma jogar?
() Somente os Jogos Educativos () Jogos do Cartoon Network () Jogos da Barbie
() Jogos de Corrida () Outros, favor especificar alguns _____
8. Com qual frequência você usa o computador?
() Todos os dias () Um a três vezes por semana () Somente na escola
9. Você prefere jogar sozinho ou com os amigos?
() Sozinho - Porque _____ () Com amigos – Porque _____

Jogos de videogame

10. O que você acha dos jogos de videogame? () Muito legal () Pouco legal () Detesto
11. Tem videogame em casa? () Sim () Não
12. Você joga jogos de videogame? () Sim () Não

Falando do Quarto 3D

13. Você considera que o jogo tem qual nível de comunicação entre jogadores, usando a caixa de texto para enviar mensagens à outros jogadores e movimentos das peças ?

- Fácil Médio Difícil

14. Você considera que a jogabilidade do Jogo com RV é:

- Boa Regular Ruim

15. Utilizar o mouse para movimentar as peças colocando-as no tabuleiro foi?

- Fácil Médio Difícil

16. Para você usar este *software* foi?

- Fácil Médio Difícil

Experiências com o jogo

17. Entender as regras do jogo foi?

- Fácil Pouco Fácil Difícil

18. Você identificou os atributos (características) das peças?

- Sim, todos Não Alguns

19. Identificar os atributos das peças foi importante para você realizar as jogadas?

- Sim Não Um pouco

20. O que você acha de aprender enquanto joga no computador?

- Bom Muito bom Chato

21. Você considera o Quarto 3D:

- Interessante Pouco Interessante Chato

1. Este espaço foi reservado para observações e sugestões.

ANEXO B – Quadros demonstrativos. Resultados da avaliação geral.

Sexo

		Frequência	%	% Válido
Válido	Masculino	15	44,0	44,0
	Feminino	19	56,0	56,0
Em branco				
	Total	34	100,0	100,0

1) Tem computador em casa?

		Frequência	%	% Válido
Válido	Sim	29	85,2	87,9
	Não	4	11,8	12,1
Em branco		1	3,0	100,0
	Total	34	100,0	

2) Tem Internet em casa?

		Frequência	%	% Válido
Válido	Sim	24	70,6	75,0
	Não	8	23,5	25,0
	Total	32	94,1	100,0
Em branco		2	5,9	
	Total	34	100,0	

3) Você costuma usar o computador para fazer o quê?

		Frequência	%	% Válido
Válido	Para jogar	6	17,6	17,6
	Fazer pesquisas	2	5,9	5,9
	Para brincar	5	14,7	14,7
	Mais de uma opção	17	50,0	50,0
	Outros	4	11,8	11,8
	Total	34	100,0	
Em branco				
Total		34	100,0	100,0

4) Você joga no computador?

		Frequência	%	% Válido
Válido	Sim	26	76,4	76,4
	Não	8	23,6	23,6
Em branco				
	Total	34	100,0	100,0

5) Quanto tempo você costumar jogar ?

		Frequência	%	% Válido
Válido	1- 2hs	32	94,12	94,12
	2-4hs	2	5,88	5,88
Em branco				
	Total	34	100,0	100,0

6) Você joga pela Internet com seus amigos ao mesmo tempo?

		Frequência	%	% Válido
Válido	Sim	12	35,2	35,2
	Não	22	64,8	64,8
Em branco				
	Total	34	100,0	100,0

7) Que tipos de jogos você costuma jogar ?

		Frequência	%	% Válido
Válido	Jogos da educativos	4	11,7	12,1
	Jogos do Cartoon Network	2	5,8	6,0
	Jogos da Barbie	3	8,9	9,2
	Jogos de corrida	1	3,0	3,0
	Outros	6	17,6	18,1
	Mais de uma opção	17	50,0	51,6
	Total	33	97,00	100,00
Em branco		1	3,0	
Total		34	100,0	

8) Com qual frequência você usa o computador?

		Frequência	%	% Válido
Válido	Todos os dias	21	61,8	65,7
	Um ou duas vezes por semana	7	20,5	21,8
	Somente na escola	4	11,8	12,5
	Total	32	94,1	100,00
Em branco		2	5,9	
Total		34	100,0	

9) Você prefere jogar sozinho ou com os amigos?

		Frequência	%	% Válido
Válido	Sozinho	9	26,4	28,0
	Com amigos	23	67,7	72,0
	Total	32	94,1	100,00
Em branco		2	5,9	
Total		34	100,0	

10) O que você acha dos videogame?

		Frequência	%	% Válido
Válido	Muito Legal	27	79,4	79,4
	Pouco Legal	5	14,7	14,7
	Detesto	2	5,9	5,9
	Total	34	100,0	
Em branco				
Total			100,0	100,0

11) Você tem videogame em casa?

		Frequência	%	% Válido
Válido	Sim	26	76,4	76,4
	Não	8	23,6	23,6
Em branco				
	Total	34	100,0	100,0

12) Você joga jogos de videogame?

		Freqüência	%	% Válido
Válido	Sim	28	82,3	84,9
	Não	5	14,7	15,1
	Total	34	97,0	100,00
Em branco		1	3,0	
Total		34	100,0	

13) Você considera que o jogo tem qual nível de comunicação entre os jogadores?

		Freqüência	%	% Válido
Válido	Fácil	23	67,6	67,6
	Médio	10	29,5	29,5
	Difícil	1	2,9	2,9
	Total	34		
Em branco				
Total		34	100,0	100,00

14) Como você considera a jogabilidade com RV?

		Freqüência	%	% V Válido
Válido	Boa	29	85,3	85,3
	Regular	4	11,8	11,8
	Ruim	1	2,9	2,9
	Total	34		
Em branco				
Total		34	100,0	100,00

15) Utilizar o mouse para movimentar as peças colocando-as no tabuleiro foi?

		Freqüência	%	% Válido
Válido	Fácil	25	73,5	73,5
	Médio	7	20,6	20,6
	Difícil	2	5,9	5,9
	Total	34	100,00	100,00
Em branco				
Total		34	100,0	

16) Para você usar este *software* foi ?

		Frequência	%	% Válido
Válido	Fácil	27	79,5	79,5
	Médio	6	17,6	17,6
	Difícil	1	2,9	2,9
	Total	34		
Em branco				
Total		34	100,0	100,00

17) Entender as regras do jogo foi ?

		Frequência	%	% Válido
Válido	Fácil	17	50,0	51,6
	Médio	11	32,3	33,3
	Difícil	5	14,7	15,1
	Total	33	97,0	100,00
Em branco		1	3,0	
Total		34	100,0	

18) Você identificou os atributos (características) das peças?

		Frequência	%	% Válido
Válido	Sim, todos	32	94,2	94,1
	Não	1	2,9	2,9
	Alguns	1	2,9	2,9
	Total	34		
Em branco				
	Total	34	100,0	100,00

19) Identificar os atributos das peças foi importante para você realizar as jogadas?

		Frequência	%	% Válido
Válido	Sim	32	94,2	94,1
	Não	1	2,9	2,9
	Um pouco	1	2,9	2,9
	Total	34		
Em branco				
Total		34	34	100,0

20) O que você acha de aprender enquanto joga no computador?

		Freqüência	%	%Válido
Válido	Muito Bom	26	76,5	76,5
	bom	3	8,8	8,8
	Chato	5	14,7	14,7
	Total	34		
Em branco				
Total		34	100,0	100,00

21) Você considera o Quarto 3D:

		Freqüência	%	%Válido
Válido	Interessante	33	97,1	97,1
	Pouco Interessante	1	2,9	2,9
	Chato	0	0,0	0,0
	Total	34		
Em branco				
Total		34	100,0	100,00