

VICENTE PIMENTEL DE SAMPAIO GÓES

**FADAS: UM SISTEMA DE AUTORIA PARA CRIAÇÃO E
MANIPULAÇÃO DE MUNDOS VIRTUAIS
EM AMBIENTES IMERSIVOS**

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do Título de Mestre em
Engenharia.

São Paulo
2003

VICENTE PIMENTEL DE SAMPAIO GÓES

**FADAS: UM SISTEMA DE AUTORIA PARA CRIAÇÃO E
MANIPULAÇÃO DE MUNDOS VIRTUAIS
EM AMBIENTES IMERSIVOS**

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do Título de Mestre em
Engenharia.

Área de Concentração:
Sistemas Eletrônicos

Orientadora:
Profa. Dra. Roseli de Deus Lopes

São Paulo
2003

Góes, Vicente Pimentel de Sampaio

**FADAS: um sistema de autoria para criação e manipulação
de mundos virtuais em ambientes imersivos
145 p.**

**Dissertação (Mestrado) – Escola Politécnica da
Universidade de São Paulo. Departamento de Engenharia
de Sistemas Eletônicos.**

**1. Realidade Virtual 2. Linguagem de Programação 3.
Ensino por computador
I. Universidade de São Paulo. Escola Politécnica. Departamento
de Engenharia de Sistemas Eletrônicos II.t**

Dedico esta dissertação à minha esposa Rose e ao meu filho Vicente pelo apoio, colaboração, incentivo e sacrifícios que tornaram possível esta pesquisa.

AGRADECIMENTOS

Aos amigos e orientadores Profa. Dra. Roseli de Deus Lopes e Prof. Dr. Marcelo Knörich Zuffo.

À Profa. Dra. Maria Zilda da Cunha pelo apoio e colaboração.

Às crianças e seus pais que permitiram a colaboração nesta pesquisa.

Aos colegas pesquisadores e desenvolvedores do Laboratório de Sistemas Integráveis, InfoSERVER Informática e BOVESPA.

À minha avó Edna Vieira Pimentel e ao meu pai Vicente Sampaio Góes Neto.

À minha família Sguilaro pelo apoio.

À minha mãe Maria Lúcia Pimentel de Sampaio Góes pela colaboração e pelo amor à pesquisa, à educação e à criança que me inspiraram desde pequeno.

Ao meu filho Vicente Pimentel de Sampaio Góes Junior, pela colaboração e entusiasmo pelo projeto.

À minha esposa Rosemary Sguilaro Sampaio Góes pela colaboração e sacrifícios que tornaram possível esta pesquisa.

A todos que, direta ou indiretamente, colaboraram na execução deste trabalho.

RESUMO

Os avanços tecnológicos trouxeram novos recursos e meios pelos quais o homem pode se expressar de maneiras diferentes, através de linguagens visuais, sonoras, gestuais, tácteis, e outras, que se complementam. Estes meios abrem um leque maior de oportunidades para o homem usar a sua capacidade criativa, explorar mundos, conceitos e experiências, aprender e construir conhecimento. Porém esta tecnologia ainda é pouco ou mal utilizada nos estabelecimentos de ensino neste país e no resto do mundo. A tecnologia existe e avança a cada momento, mas ainda se encontra dificuldade em construir e manipular mundos virtuais em ambientes imersivos com múltiplas projeções. Este trabalho propõe o sistema FADAS, um sistema de autoria para ambientes imersivos de múltiplas projeções, para estudantes, professores e pesquisadores utilizarem a Realidade Virtual como mecanismo facilitador e enriquecedor do processo de aprendizagem e pesquisa. Por meio de várias tecnologias, fornece uma ferramenta, tanto para o “usuário-autor”, que constrói cenários virtuais, baseados ou não na realidade, quanto para o “usuário-explorador”, que navega e interage com os mundos virtuais, criando suas próprias experiências. O sistema FADAS pode ser utilizado em ambiente imersivo de múltiplas projeções estereoscópicas, disponibilizando uma linguagem de programação, que encapsula as complexidades das tecnologias envolvidas, e permite a criação e manipulação de mundos virtuais por usuários não especialistas.

ABSTRACT

Technological advances brought new resources and means where man can express himself in several ways, through languages of several types: visual, sonorous, gesture, tactile, and others that complement each other. These means open up a bigger span of opportunities for man to use his creative capacity, to explore worlds, concepts and experiences, learn and build up knowledge. Nevertheless this technology is still limited or badly used in education in this country and in the rest of the world. Technology exists and advances at every moment, but there is still difficulty in building up and manipulating virtual worlds in multiprojection immersive environments. This work proposes the FADAS system, an authorship system for multiprojection immersive environments, for students, teachers and researches to use Virtual Reality as a facilitating and enriching mechanism of the learning process and research. Through several technologies, it provides a tool to build up scenarios, based, or not, in real worlds, and to explore, navigate and interact with them, creating their own experiences. The FADAS system can be used in a stereo immersive multiprojection environment. The system provides a programming language, which encapsulates technology complexities, allowing users, who are not specialists, to create and manipulate virtual worlds.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Objetivo	3
1.2	Relevância	3
1.3	Motivação	3
1.3	Estrutura da Dissertação	4
2	FUNDAMENTAÇÃO TEÓRICA	5
2.1	O Computador na Educação	6
2.1.1	O agente de aprendizagem	12
2.1.2	Softwares educacionais e de entretenimento	14
2.2	Ambientes Virtuais Imersivos	16
2.2.1	Realidade Virtual Aplicada à Educação	17
2.2.2	Interação	20
2.2.3	Rastreamento	22
2.2.4	Estereoscopia	23
2.2.4	Som tridimensional	25
2.2.5	Animação	26
2.2.6	Múltiplas Projeções	27
2.3	Trabalho Colaborativo	29
2.4	Linguagens de Programação e Programas Tradutores	33
2.5	Exemplos de Ambientes Virtuais de Autoria	34
2.5.1	SuperLogo	34
2.5.2	Imagine	36
2.5.3	Alice	37
3	PROPOSTA DE UM SISTEMA DE AUTORIA PARA CRIAÇÃO DE AMBIENTES IMERSIVOS: O SISTEMA FADAS	40
3.1	Linguagem Fadas	44
3.2	Manipulação de Objetos Tridimensionais	46
3.2.1	Navegação em Ambiente Imersivo	46
3.2.2	Manipulação de Objetos em Ambiente Imersivo	47
3.3	Criação de Objetos Tridimensionais	47
3.3.1	Hierarquia entre Objetos	48
3.3.2	Biblioteca de Objetos e Sons	48
3.3.3	Criação de Pontos, Linhas, Polígonos, Sólidos e Textos Tridimensionais	49
3.3.4	Criação de Fontes de Luz e Som Tridimensional	49
3.4	Colaboração	49
3.5	Comparação entre os ambientes virtuais de autoria	50
4	IMPLEMENTAÇÃO	52
4.1	Arquitetura	53
4.2	Classes Internas	55
4.3	Hierarquia entre objetos	57
4.4	Plataforma de Desenvolvimento	58

4.4.1	Hardware	58
4.4.2	<i>Software</i>	58
4.5	Interfaces com o usuário	59
4.6	Linguagem Fadas	63
4.6.1	Sintaxe, Semântica e Regras Propostas	64
4.6.1.1	Programa	64
4.6.1.2	Declaração de Caixa.....	65
4.6.1.3	Ação de um Objeto.....	65
4.6.1.4	Atribuição	66
4.6.1.5	Comando Condicional	67
4.6.1.6	Comando em laço	68
4.6.1.7	Rotina criada pelo usuário	68
4.6.1.7	Comentários.....	69
4.6.1.8	Eventos	69
4.6.2	Exemplos	70
4.6.2.1	Movimentação	70
4.6.2.2	Rotina criada pelo usuário	71
4.6.2.3	Codificando Eventos	72
4.7	Interpretador.....	73
4.7.1	Reconhecedor Léxico Sintático	73
4.7.2	Analisadores Léxico, Sintático e Semântico	74
4.7.3	Tratamento de Erros	76
4.8	Biblioteca Gráfica OpenGL	76
4.9	Imagens Estereoscópicas	80
4.10	Movimento e Rotação Relativa.....	82
4.11	Navegação.....	84
4.12	Animação	85
4.12.1	Animação baseada em script	85
4.12.2	Animação baseada em keyframe	86
4.13	Detecção de Colisão entre Objetos	87
4.14	Ciclo Principal da Aplicação Mundo Virtual.....	88
4.15	Arquivos do Mundo Virtual	88
4.16	Som	90
4.17	Comunicação entre processos	90
4.18	Multi-projeção e Rastreamento do usuário.....	91
4.19	Trabalho Colaborativo	93
5	TESTES E AVALIAÇÃO DOS RESULTADOS	95
5.1	Objetivo	97
5.2	Roteiro	97
5.3	Atividades.....	98
5.4	Depoimentos	99
5.5	Resultados.....	102
5.6	Conclusão.....	102

6	CONCLUSÕES E TRABALHOS FUTUROS	103
6.1	Contribuições	104
6.2	Trabalhos Futuros	105
6.3	Conclusão.....	106
	ANEXO A - COMANDOS DO SISTEMA FADAS	107
	ANEXO B - TIPOS DE EVENTO DO SISTEMA FADAS	137
7	LISTA DE REFERÊNCIAS.....	139

APÊNDICE I – Gramática para Reconhecimento Léxico da Linguagem Fadas

APÊNDICE II – Gramática dos Comandos Básicos da Linguagem Fadas

APÊNDICE III – Gramática da Linguagem Fadas

APÊNDICE IV – Mensagens de Erro do Sistema FADAS

APÊNDICE V – Configuração do Sistema FADAS

LISTA DE FIGURAS

Fig. 1 – Disparidade das imagens na retina (STEREOGRAPHICS, 1997)	23
Fig. 2 – Imagens estereoscópicas por computador (STEREOGRAPHICS, 1997)	24
Fig. 3 – Paralaxes positiva, zero e negativa (STEREOGRAPHICS, 1997).....	24
Fig. 4 – Ambiente imersivo com alto-falantes (NAEF; STAADT; GROSS 2002) ...	26
Fig. 5 – Projeções em um CAVE (CRUZ-NEIRA et al., 1992)	28
Fig. 6 – <i>Layout</i> da CAVERNA Digital (ZUFFO et al., 2001)	28
Fig. 7 – Usuários e operadores na CAVERNA Digital (ZUFFO et al., 2001).....	29
Fig. 8 – Arquitetura Centralizada (SUTHERS, 2001)	32
Fig. 9 – Arquitetura Replicada (SUTHERS, 2001).....	32
Fig. 10 – Arquitetura Distribuída (SUTHERS, 2001).....	32
Fig. 11 – Arquitetura Híbrida (SUTHERS, 2001)	33
Fig. 12 – Exemplo de comandos no ambiente SuperLogo	35
Fig. 13 – Exemplo de projeto no ambiente Imagine	36
Fig. 14 – Exemplo de tela construída no ambiente Alice 99.....	38
Fig. 15 – Tela de abertura do protótipo do sistema FADAS	44
Fig. 16 – Telas do sistema FADAS: Mundo Virtual e Console	45
Fig. 17 – Arquitetura do sistema FADAS	54
Fig. 18a – Classes internas do sistema FADAS	55
Fig. 18b – Classes internas do sistema FADAS	56
Fig. 19 – Tela de comandos do console	60
Fig. 20 – Janela Objetos	61
Fig. 21 – Janela Cor.....	61
Fig. 22 – Janela “Animações”	62
Fig. 23 – Janela “Sons”	62
Fig. 24 – Janela “Câmeras”	63
Fig. 25 – Janela “Fontes de Luz”	63
Fig. 26 – Gerador de Reconhecedor Léxico Sintático	73
Fig. 27 – Volume de Visualização	77
Fig. 28 – Modelo de câmeras com eixos paralelos (STEREOGRAPHICS, 1997)	81
Fig. 29 – Translação horizontal da imagem (HIT) (STEREOGRAPHICS, 1997)	82
Fig. 30 - Sistema de coordenadas global.....	82
Fig. 31 – Duas projeções sincronizadas do sistema FADAS	92
Fig. 32 – Observador no centro da CAVERNA Digital.....	92
Fig. 33 – Observador andando na CAVERNA Digital.....	93
Fig. 34 – Primeiros usuários testando o protótipo do sistema FADAS	96
Fig. 35 – Pesquisadores entrevistando um usuário	96
Fig. 36 – Usuários interagindo com a CAVERNA Digital.....	98

LISTA DE TABELAS

Tabela I – Comparação entre ambientes virtuais de autoria	51
Tabela II – Navegação com teclas.....	84
Tabela III – Navegação com <i>joystick</i>	85
Tabela IV – Síntese dos comandos do sistema FADAS	135

LISTA DE EXEMPLOS DE CÓDIGO

Exemplo de código 1 – Hierarquia entre objetos (linguagem C++)	58
Exemplo de código 2 – Sintaxe Programa (linguagem Fadas)	64
Exemplo de código 3 – Declaração de Caixas (linguagem Fadas)	65
Exemplo de código 4 – Ação de um objeto (linguagem Fadas).....	66
Exemplo de código 5 – Atribuição (linguagem Fadas).....	67
Exemplo de código 6 – Condição (linguagem Fadas).....	67
Exemplo de código 7 – Comparação (linguagem Fadas).....	68
Exemplo de código 8 – Laço (linguagem Fadas)	68
Exemplo de código 9 – Declaração de Rotina (linguagem Fadas)	69
Exemplo de código 10 – Chamada de Rotina (linguagem Fadas)	69
Exemplo de código 11 – Declaração de Evento (linguagem Fadas).....	70
Exemplo de código 12 – Movimento 1 (linguagem Fadas).....	70
Exemplo de código 13 – Movimento 2 (linguagem Fadas).....	71
Exemplo de código 14 – Rotina (linguagem Fadas).....	71
Exemplo de código 16 – Reconhecedor (linguagem C++)	74
Exemplo de código 17 – Interpretador (linguagem C++)	75
Exemplo de código 18 – Cor (linguagem Fadas).....	79
Exemplo de código 19 – Luz Emissiva (linguagem Fadas).....	79
Exemplo de código 20 – Opacidade (linguagem Fadas).....	79
Exemplo de código 21 – Texto tridimensional (linguagem Fadas)	86
Exemplo de código 22 - Animação baseada em script (linguagem Fadas).....	86
Exemplo de código 23 – Animação baseada em keyframe (linguagem Fadas).....	87
Exemplo de código 24 – Detecção de colisão (linguagem C++)	88
Exemplo de código 25 – Estrutura de arquivo de objeto 3D (linguagem C++).....	89
Exemplo de código 26 – Som (linguagem Fadas).....	90
Exemplo de código 27 – Som 3D (linguagem Fadas).....	90

LISTA DE ABREVIATURAS E SIGLAS

API.....	<i>Application Program Interface</i> Interface de Programa de Aplicação
ASCII.....	<i>American Standard Code for Information Interchange</i> Padrão Americano de Codificação para Troca de Informação
HIT.....	<i>Horizontal Image Translation</i> Translação Horizontal de Imagem
LSI.....	Laboratório de Sistemas Integráveis
MCI.....	<i>Media Control Interface</i> Interface de Controle de Mídia
MIDI.....	<i>Musical Instrument Digital Interface</i> Interface Digital de Instrumento Musical
NICE	<i>Narrative-based Immersive Constructionist/ Collaborative</i> <i>Environments</i> Ambientes Construcionistas / Colaborativos Imersivos baseados em Narrativa
NIED.....	Núcleo de Informática Aplicada à Educação
RV.....	Realidade Virtual
TCP/IP.....	<i>Transmission Control Protocol/Internet Protocol</i>
UDP.....	<i>User Datagram Protocol</i>
ZPS.....	<i>Zero Parallax Setting</i> Configuração de Paralaxe Zero
2D.....	2 dimensões
3D.....	3 dimensões

CAPÍTULO 1

INTRODUÇÃO

*“Naquela noite, Lili ligou Ulisses para chamar, na tela, as imagens da mãe sempre sorridente. O encontro tão incomum ia consolar a solidão da menina quando um detestável raio caiu perto do prédio dos Pena. A forte explosão deu um grande susto em Lili-a-Desastrada. Será que seu dedo esbarrou numa tecla errada ? Ou será que caiu a corrente elétrica ? Fosse o que fosse, o fato é que sua mãe tinha sumido da tela, sumido de sua vida. Por mais que Lili acionasse todos os comandos do teclado, não conseguia encontrar os arquivos onde a mãe estava! Será que Lúcia tinha sido apagada do disco rígido de Ulisses ? A menina acabava de deletar a mãe. Ela desapareceu!
- Mãe... Mamãe, repetia, aflita, Lili.”*

Alexandre Jardin¹

¹ JARDIN, A. *Cibermãe: uma viagem extraordinária dentro do computador*. Trad. de Estela dos Santos Abreu. São Paulo: Moderna, 1998.

O avanço da Tecnologia de Informação, em especial da Realidade Virtual (RV), traz novos meios para o homem se expressar, comunicar, criar, explorar, aprender e construir conhecimento. Esta pesquisa, através da concepção e desenvolvimento do sistema FADAS, explora recursos tecnológicos para a criação e manipulação de mundos virtuais em ambientes imersivos.

A Realidade Virtual compreende o conjunto de conhecimentos e tecnologias relacionadas com possibilidade de imersão do usuário em ambientes simulados pelo computador. Particularmente a imersão é oferecida através de sistemas avançados de exibição de imagens e sistemas complementares a outros estímulos perceptuais ao ser humano como o som e eventualmente o tato. Várias áreas da engenharia (aeronáutica, automobilismo e exploração petrolífera), meteorologia e atividades críticas (simuladores médicos, simuladores militares), têm utilizado com sucesso a Realidade Virtual no processo de análise, entendimento e tomada de decisões.

Recentemente com o avanço das inovações da informática e a Lei de Moore^{*}, sistemas computacionais pessoais de alto desempenho têm sido disponibilizados a custos acessíveis, este fato tem acelerado de forma significativa a utilização massificada da Realidade Virtual em áreas de forte relevância humana e social como a saúde, educação, entretenimento e cultura.

Este trabalho busca identificar as tecnologias disponíveis, e propõe remover suas complexidades para que o pesquisador, não especialista, possa empregá-las em diversas áreas, em especial, na educação, onde o computador está adquirindo papéis mais importantes.

O sistema FADAS, por meio de várias tecnologias, fornece uma ferramenta, tanto para o “usuário-autor”, que constrói cenários virtuais, baseados ou não na realidade, quanto para o “usuário-explorador”, que navega e interage com os mundos virtuais, criando suas próprias experiências.

^{*} Lei empírica que estabelece que o número de dispositivos num circuito integrado dobra a cada 18 meses (ZUFFO, 2003)

1.1 Objetivo

O objetivo principal deste trabalho é permitir a construção e manipulação de mundos virtuais em ambientes imersivos com múltiplas projeções, da maneira mais simples possível, para tornar estes ambientes acessíveis a usuários não especialistas e possibilitar o seu emprego em aplicações educacionais.

1.2 Relevância

A universidade, nesta era da Tecnologia da Informação, tem um papel importante para apresentar os meios necessários para a sociedade acompanhar e explorar bem os novos recursos, em especial, iluminar os caminhos para a escola utilizar os recursos computacionais de forma a enriquecer o processo de aprendizagem.

Particularmente acredito que a disponibilidade de ferramentas de autoria simples, práticas e de fácil uso e acesso, são um passo importante na universalização do uso efetivo da Realidade Virtual na educação escolar.

Este trabalho procura contribuir neste papel, tornando a Realidade Virtual mais acessível, mais próxima, mesmo para aqueles não especialistas na área, como por exemplo, um professor de ensino fundamental ou médio e seus alunos.

Um aspecto inovador deste projeto é a disponibilidade destas ferramentas de autoria em ambientes de Realidade Virtual Imersiva e também a disponibilidade destas ferramentas em computadores pessoais de baixo custo.

1.3 Motivação

A disponibilidade de ambientes educacionais imersivos é ainda objeto de pesquisa por vários grupos no mundo. Observa-se também poucas publicações nesta área de Realidade Virtual e imersão na literatura especializada em novas tecnologias na educação.

Particularmente o grupo de pesquisa onde estou inserido, desenvolveu e implementou o primeiro sistema de Realidade Virtual totalmente imersivo da América Latina denominado CAVERNA Digital, apresentada com detalhes no capítulo 2.

Uma necessidade observada de forma patente e imediata na CAVERNA Digital foi a falta de aplicações da Realidade Virtual na educação. Constatamos que esta necessidade é maior ainda no público jovem, em contraponto à abundância de jogos e simuladores voltados ao entretenimento. No capítulo 2 discorreremos com profundidade a respeito do assunto.

Finalmente possuo uma forte motivação pessoal na direção da facilitação do uso e prática da informática de forma natural às crianças no ambiente escolar.

1.3 Estrutura da Dissertação

Esta dissertação é composta por seis capítulos. O primeiro capítulo é uma breve introdução ao trabalho, destacando-se os objetivos, relevância e motivação que levaram a realização desta pesquisa.

O segundo capítulo apresenta a fundamentação teórica, abordando tanto aspectos tecnológicos quanto a sua aplicabilidade na educação. Neste capítulo uma revisão profunda da literatura em novas tecnologias na educação é realizada com um foco específico na área de Realidade Virtual imersiva.

O terceiro capítulo, apresenta a concepção do FADAS, que é um sistema de autoria que permite a criação e manipulação de mundos virtuais em ambientes imersivos de múltiplas projeções. Neste capítulo, descrevemos com detalhes as características funcionais do sistema FADAS.

O capítulo quatro descreve o projeto e implementação do sistema FADAS, com especial destaque aos aspectos técnicos relevantes a esta pesquisa.

O capítulo cinco apresenta uma avaliação experimental preliminar do sistema FADAS, no caso o sistema foi instalado na CAVERNA Digital e utilizado por um grupo de usuários infantis (idade de sete a doze anos), esta avaliação experimental foi acompanhada por educadores.

O capítulo seis conclui o trabalho apresentando as contribuições e sugestões para trabalhos futuros.

CAPÍTULO 2

FUNDAMENTAÇÃO TEÓRICA

*“Foi o velho pastor que me explicou a razão de sua serenidade:
 – Garoto, de manhã, os animais sempre sobem. Vão para os cumes.
 Não que o pasto de lá seja sempre mais abundante ou mais fácil; mas um
 instinto de se lançar os braços para o azul do céu e partir ao assalto dos
 cumes. O capim, conquistado à força de músculos e tenacidade,
 tem um valor exaltante, talvez só por ter sido muito desejado...
 – Pode ficar tranqüilo: vamos encontrá-los todos lá em cima! E acrescentou: -
 Só me preocupa o pequeno bando de Léon, domesticado demais, habituado
 demais a comer nos pastos e na manjedoura, e que tem como que a nostalgia
 das barreiras e do estábulo. Parece que já não têm força para subir; o ideal
 deles já não é lá em cima, mas embaixo... Preferem a rédea ao azul do céu...
 Já não são ovelhas dignas e orgulhosas: são cães!
 Ouça os chocalhos, lá em cima, diante de nós! Só a noitinha, quando o sol
 se extinguir por trás do Rocheroux, os nossos animais também vão descer para
 a calma e segurança do vale, para amanhã voltarem a subir mais alto ainda.”*

*Freinet**

*“E as crianças – diria o pastor – são como as ovelhas: querem subir sempre.
 Você só terá paz e certeza se souber ajudá-las, às vezes precedê-las na subida
 aos cumes, ou segui-las... Infelizes dos seres domesticados cedo demais, que
 perderam o sentido da subida e que, como velhos em fim de corrida, preferem,
 ao ar do espaço e ao azul do céu, a coleira da sujeição e ração da renúncia.
 São bons todos os caminhos que levam para as alturas.”*

*Lúcia Pimentel Góes***

* FREINET, C. *Pedagogia do Bom Senso*. Trad. de J. Baptista. 2.ed. São Paulo: Martins Fontes, 1988.

** GÓES, L.P. *A Aventura da Literatura para Crianças*. 5.ed. São Paulo: Melhoramentos, 1993.

Este capítulo apresenta a fundamentação teórica necessária para o tema desta dissertação, abordando tanto aspectos tecnológicos quanto de aplicabilidade na educação. Abrange o uso do computador na educação, as tecnologias de Realidade Virtual imersiva e sua aplicação na aprendizagem, conceitos de trabalho colaborativo e de linguagem de programação, finalizando com alguns exemplos de ambientes virtuais de autoria.

2.1 O Computador na Educação

Atualmente, o cidadão se depara cada vez mais com a informática, muitas vezes sem notar. Ela pode ser encontrada nos mais diversos lugares. O ritmo de sua evolução é alucinante, com as capacidades de processamento, armazenagem e comunicação aumentando e os custos diminuindo. Conforme Zuffo (2003), vivemos na “Infoera, a era da informação e do conhecimento”, onde “o conhecimento e a educação ocupam papel central no processo de desenvolvimento e progresso social”.

Piaget apud Papert (1999b) coloca que as crianças têm real entendimento, somente daquilo que elas mesmas inventam e, cada vez que tentamos ensiná-las algo muito rapidamente, não permitimos a reinvenção por elas mesmas. Piaget apud Papert (1999b) constatou que as crianças não são vasos vazios a serem encheidos de conhecimentos, mas sim, construtores de conhecimento, pequenos cientistas que estão constantemente criando e testando suas próprias teorias sobre o mundo. Papert (1999b), constatando que a tecnologia digital dá às crianças muito mais autonomia para explorar mundos maiores, alerta pais e educadores de que as idéias pioneiras de Piaget se tornaram mais urgentes.

Gardner (1995) define inteligência: “a capacidade de resolver problemas ou de elaborar produtos que sejam valorizados em um ou mais ambientes culturais ou comunitários”. O pesquisador destaca a pluralidade do intelecto ao identificar sete inteligências: inteligência lingüística, inteligência lógico-matemática, inteligência espacial, inteligência musical, inteligência corporal-cinestésica, inteligência interpessoal e inteligência intrapessoal. Gardner considera as inteligências como potenciais puros, biológicos e que o propósito da escola deveria ser o de desenvolver as inteligências e ajudar as pessoas a atingirem objetivos de ocupação e de passatempo adequados ao seu espectro particular de inteligências.

Para Gardner (1995), a escola ideal do futuro seria centrada no indivíduo, pois nem todas as pessoas têm os mesmos interesses e habilidades, nem todos aprendem da mesma maneira e ninguém pode aprender tudo o que há para ser aprendido. Segundo o pesquisador, as novas tecnologias podem facilitar a implementação desta escola centrada no indivíduo.

Segundo Freire (2000): “As crianças precisam crescer no exercício desta capacidade de pensar, de indagar-se e de indagar, de duvidar, de experimentar hipóteses de ação”

O computador, é claro, já chegou e está cada vez mais presente no trabalho, na escola e na residência. Mas na escola, ele aparece, muitas vezes, segundo Valente (1999), sem modificar a forma tradicional de ensino, apenas reforçando o processo instrucionista. Nestes casos, ele é usado para continuar transmitindo a informação para o aluno. Porém, o mesmo teórico coloca que as possibilidades de utilização de computadores na educação são muito mais diversificadas, interessantes e desafiadoras, podendo enriquecer ambientes de aprendizagem e auxiliar o aprendiz no processo de construção de seu conhecimento.

Os avanços tecnológicos podem e devem contribuir para avanços na educação. Segundo Freire (2000): “A mudança é uma constatação natural da cultura e da história. O que ocorre é que há etapas, nas culturas, em que mudanças se dão de maneira acelerada. É o que se verifica hoje. As revoluções tecnológicas encurtam o tempo entre uma e outra mudança”.

No Brasil, Valente (1999) relata que desde 1982, as políticas de implantação da informática na escola pública, têm sido norteadas na direção da mudança pedagógica. Isto foi possível porque as propostas pedagógicas foram fundamentadas nas pesquisas realizadas com universidades e escolas da rede pública, como por exemplo, o projeto EDUCOM (NIED, 1983); (NIED, 1986); (VALENTE, 1999) realizado com as seguintes universidades: UFPe (Universidade Federal de Pernambuco), UFMG (Universidade Federal de Minas Gerais), UFRJ (Universidade Federal do Rio de Janeiro), UFRGS (Universidade Federal do Rio Grande do Sul) e UNICAMP (Universidade Estadual de Campinas).

O programa educacional PROINFO (1997), iniciado em 1997 no âmbito da Secretaria de Educação a Distância do Ministério da Educação, é outro exemplo

desta política, introduzindo as novas tecnologias de informática e telecomunicações no sistema escolar público levando sugestões de inovações pedagógicas e gerenciais para as escolas. O programa tem na preparação de recursos humanos sua principal condição de sucesso.

Apesar da história da Informática na Educação no Brasil ter mais de vinte anos, os resultados foram aquém do desejado e esperado (VALENTE; ALMEIDA, 1997). Um dos fatores deste insucesso, além da falta de verbas, é a preparação inadequada de professores, em vista dos objetivos de mudança pedagógica propostos. Para Valente (1999), “os resultados obtidos não foram suficientes para sensibilizar ou alterar o sistema educacional como um todo”. Diversas experiências instaladas no Brasil apresentaram mudanças pedagógicas fortemente enraizadas e produzindo frutos. No entanto, essas idéias não se alastraram e isso aconteceu, principalmente, diz Valente, pelo fato de termos subestimado as implicações das mudanças pedagógicas propostas no sistema educacional como um todo: a mudança na organização da escola, na dinâmica da sala de aula, no papel do professor e dos alunos e na relação com o conhecimento. O educador também apresenta o retrato da implantação do computador nas escolas particulares, onde o computador é usado para minimizar o analfabetismo computacional dos alunos ou automatizar os processos de transmissão da informação.

A maneira instrucionista de usar o computador pode ser verificada no texto de Vieira (1998) ao escrever que as tecnologias têm auxiliado o processo de ensino e talvez o de aprendizagem, mas o resultado tem sido pouco observável na prática e a educação formal continua essencialmente inalterada.

O computador tem sido por diversas vezes mal empregado no processo de aprendizagem da criança e do adolescente. Outro motivo desta situação é a falta de uma equipe especializada em informática para dar suporte aos coordenadores e professores, para criarem juntos experiências que enriqueçam a aprendizagem. Um ponto importante a ressaltar é que a aprendizagem do professor junto com o aluno é um momento raro mas especialmente benéfico na construção do conhecimento, de modo que, o professor que não conhece o computador e tecnologias como a Realidade Virtual, não deve se intimidar pela sua falta de conhecimento. Pelo contrário, ele não é obrigado a ter este conhecimento, e deve aproveitar para aprender

junto com o aluno. Quando ele aprende junto, sem saber quais serão os resultados de uma experiência, certamente não estará bloqueando a imaginação, criatividade e curiosidade de seus alunos, propiciando um ambiente excelente para a criação de conhecimento. O aluno, assim, aprende a aprender.

Esta situação onde o professor aprende junto com o estudante é incentivada por Papert (1999a), ao enfatizar que um dos aspectos mais importantes do espírito Logo é encorajar situações que o professor nunca tenha visto antes, levando-o a unir-se aos estudantes como autêntico aprendiz.

A filosofia Logo (NIED, 1983); (PAPERT, 1999a), criada pelos professores Seymour Papert e Marvin Minsky, e trazida ao Brasil pelo projeto Logo da UNICAMP (Universidade Estadual de Campinas), iniciado em 1973, quando pesquisadores desta entidade tiveram a oportunidade de conhecer os seus criadores no Laboratório Logo do MIT (*Massachusetts Institute of Technology*). O objetivo inicial deste projeto (NIED, 1983), foi introduzir a linguagem Logo no Brasil: “adequá-la à realidade brasileira, com base em um estudo piloto com algumas crianças, estudo este que teria por objetivo verificar como o ambiente Logo influencia a aprendizagem”.

Papert (1999a) conta que, para muitos logoístas, Logo é uma linguagem de programação somada a uma filosofia de educação, o construtivismo. Mas ele não se contenta com esta definição e, explica que na cultura Logo, fazer acontecer é muito mais do que um princípio educacional ou pedagógico. Será melhor descrito como filosofia de vida do que uma filosofia educacional. Por este motivo Papert adotou a palavra construcionismo para se referir a tudo o que tem a ver com fazer coisas e, especialmente, a ver com aprender criando, fazendo acontecer, o que inclui a idéia de aprender fazendo e indo muito mais além.

Para Vieira (1998) a introdução de recursos tecnológicos na educação deve ser acompanhada de uma sólida formação dos professores para que eles possam utilizá-los de uma forma responsável, com potencialidades pedagógicas verdadeiras, excluindo seu uso apenas como máquinas divertidas e agradáveis para passar o tempo.

Valente (1999) afirma que se atualmente novos *softwares* ampliam as possibilidades de que o professor dispõe para o uso do computador na construção do

conhecimento, eles também demandam um discernimento maior por parte do professor e, conseqüentemente, uma formação mais sólida e mais ampla.

Sem dúvida, os professores devem receber treinamento sobre o uso do computador e de *softwares* educativos, mas noções básicas sobre uma linguagem de programação e até mesmo sobre Computação Gráfica pode ser bastante benéfico para o professor que se interessar. Também é fundamental que uma equipe especializada no emprego da informática na educação dê suporte e trabalhe junto com o professor ou coordenador. A atualização tecnológica deve ser uma constante na vida do professor e da escola. As mudanças são rápidas no mundo atual, de modo que o profissional precisa estudar, pesquisar, experimentar e aprender sempre. Este pensamento vem somar-se à reflexão de Valente; Almeida (1997): “Não se trata de criar condições para o professor dominar o computador ou o *software*, mas sim de auxiliá-lo a desenvolver conhecimento sobre o próprio conteúdo e sobre como o computador pode ser integrado no desenvolvimento desse conteúdo”.

Nas experiências com Logo, Papert (1999a) relata que freqüentemente recebia a afirmativa de que os professores não eram capazes, sua educação era limitada, não eram acostumados com as idéias do Logo, eram conservadores, e assim por diante. As experiências em larga escala com projetos Logo, publicadas por Papert, confirmam que isto só é verdadeiro se a falta de capacidade não significar incapacidade por faltar oportunidade de “aprender como”. E sugerem que “aprender como”, pode requerer mais do que geralmente é oferecido – algumas horas de preparação da equipe com “treinadores” (entre aspas) de alguma empresa de computador ou de *software*.

Valente; Almeida (1997) relatam que o interesse pelo uso do Logo já não é o mesmo, que o papel do professor neste ambiente é fundamental e o preparo do professor não é trivial.

Estas experiências com Logo só reforçam o pensamento de que o professor deve receber um treinamento contínuo nas tecnologias digitais e novas ferramentas disponíveis. Insistem também que educadores criativos, sem pré-conceitos ou barreiras, devem se unir com profissionais capacitados tecnologicamente e interessados no ensino, para compartilharem suas paixões e criarem experiências que sirvam como exemplo, auxiliando os primeiros passos do novo professor, fazendo

este visualizar os benefícios que a tecnologia digital pode trazer para a educação. Como Papert diz, lápis, giz e lousa, também são tecnologias. O computador é apenas mais uma tecnologia que pode ajudar bastante o processo de aprendizagem, sendo que aos poucos passará a ser uma ferramenta tecnológica e pedagógica freqüentemente utilizada em “sala” de aula.

Vieira (1998) coloca que a tecnologia não deve ser usada como máquina de ensinar ou aprender, mas como ferramenta pedagógica para criar um ambiente interativo que proporcione ao aprendiz, diante de uma situação problema, investigar, levantar hipóteses, testá-las e refinar suas idéias iniciais, construindo assim seu próprio conhecimento.

Valente (2001) comenta que com linguagens de programação, “o aprendiz tem de *descrever* para o computador todos os passos do processo de resolução de um problema”. O computador então os executa fielmente e fornece os resultados. O aprendiz, portanto, precisa refletir sobre o que fez e depurar suas idéias. Em seguida, reinicia este ciclo de atividade descrição-execução-reflexão-depuração-descrição.

NIED (1987) relata resultados positivos: “Nos dois grupos foram observadas mudanças de atitudes quanto à independência, espírito de iniciativa, interesse e criatividade em todos os sujeitos”. NIED (1988) também relata a contribuição de *software* do tipo linguagem de programação: “verifica-se que o aprendizado de conceitos de programação tem levado ao aprofundamento de conhecimentos específicos e esses, por sua vez, têm solicitado novos conhecimentos de programação”. Em ferramentas educativas do tipo linguagem de programação, o aprendiz precisa recorrer aos seus conhecimentos, criar idéias, testar, repensar, aprender e construir. O aluno não aprende simplesmente a executar um algoritmo, mas deve criá-lo ele mesmo.

Outro aspecto positivo trazido pela cybercultura, segundo Lévy (1995), é a simulação: “As técnicas de simulação, em particular as que envolvem imagens interativas, não substituem os raciocínios humanos, mas prolongam e transformam as capacidades de imaginação e pensamento”. Lévy também destaca a vantagem de se experimentar alterações em um modelo: “A capacidade de fazer variar facilmente os parâmetros de um modelo e observar de imediato e visualmente as conseqüências dessa variação constitui-se numa verdadeira ampliação da imaginação”.

Papert (1999a) foi muito feliz ao dizer que a educação por si mesma tem duas asas, a informação e a construção. Ele conta que existe um desequilíbrio, em grande parte por causa da ausência de tecnologias adequadas, pois o lado da construção está atrasado nas escolas, recebendo um pobre segundo lugar em relação ao lado dominante da informação.

Realmente, muitos vêem a facilidade de acesso à informação destes tempos atuais como uma revolução na educação, porém não se pode desprezar o lado de construir, criar, imaginar, inventar, que verdadeiramente gera o conhecimento humano. A comunicação em massa pode levar a uma sociedade preguiçosa, quando a informação é valorizada, sem questionamentos. Felizmente existem aqueles que acreditam na criatividade e capacidade de inovação do ser humano.

2.1.1 O agente de aprendizagem

A Tecnologia da Informação, segundo Marinho (2002), tornou-se um grande desafiador da escola, dos alunos, dos professores e dos pais. O computador permite uma aprendizagem ativa e interativa, com um acesso diversificado a inúmeras fontes de informação. “A escola, principalmente por meio do professor”, não mais detém o monopólio do conhecimento e precisa assumir esse fato”. O professor é estimulado a criar atividades pedagógicas desafiadoras aos seus alunos e que estejam vinculadas às suas próprias realidades. E muitas destas atividades serão realizadas além dos muros da escola; além do tempo das aulas. Porém com a presença do professor mediador neste espaço ampliado e tempo redimensionado, no papel de orientador, facilitador e estimulador.

O papel do agente de aprendizagem, seja ele real (um professor) ou virtual (um avatar, por exemplo), que acompanha os alunos em uma experiência com o computador, em um mundo virtual, é bastante discutida. Será que ele deve agir como facilitador? Ajudar o aprendiz quando solicitado? Como e quando deve intervir?

A interação aluno-computador, segundo Valente (2001), necessita da intervenção de um profissional, um agente de aprendizagem. Ele pode ser o professor, porém deve entender as idéias do aprendiz e saber como atuar no processo de construção do conhecimento.

Segundo Valente (1993), em ambiente Logo, o agente precisa ser um mediador que entende, que compreende, as idéias do aluno e que intervém apropriadamente na situação de modo a ser efetivo e contribuir para que o aluno compreenda o problema em questão e desenvolva suas soluções. O como intervir tem sido o maior desafio dos profissionais que trabalham com o Logo.

Segundo Papert (1999a) o espírito logoísta vê numa situação onde “deu errado” uma oportunidade de ganhar melhor entendimento daquilo que alguém está tentando fazer. Considera má prática as respostas do tipo “isto está certo” ou “isto está errado”.

Em experiências do autor em salas de aula em uma escola de informática de São Paulo, onde se ensinava uma das linguagens de programação, em determinado módulo, rapidamente se percebia um erro comum entre a maioria dos alunos. No caso a montagem dinâmica de uma cadeia de caracteres. Ao invés de corrigir o código gerado pelo aluno, ou avisá-lo com antecedência para não errar, verificou-se que a aprendizagem ocorria com êxito e com maior satisfação do aluno, quando permitia-se que o erro acontecesse e se instigava o aluno a procurá-lo, depurando o programa passo a passo se necessário. Este tipo de intervenção não mostrava o caminho correto, era mais um acompanhamento com incentivo a mais explorações e testes para que o aprendiz construísse o seu conhecimento. Quantas vezes se ouve reclamações de alunos de informática dizendo nada ter aprendido porquê o instrutor resolvera todos os exercícios do laboratório. Infelizmente, isto é um fato comum em escolas de informática, onde algumas vezes, instrutores pouco preparados pedagogicamente e por isso pouco preocupados com a aprendizagem do aluno, ministram cursos.

Uma recomendação valiosa para o agente de aprendizagem e que também deve ser empregada na educação com o computador é ensinada por Góes (1996), para o acompanhamento de um adulto durante a leitura de um livro infantil ou juvenil: ao guia-iluminador exige-se um requisito. Não ser um guia comportado, nem querer fazer do aprendiz um ser comportado.

2.1.2 *Softwares* educacionais e de entretenimento

Um dos pré-requisitos para um *software* educativo para crianças é que ele deve ser prazeroso e divertido. Druin (1999) coloca que de acordo com pesquisas realizadas nos laboratórios da Microsoft Corporation, a capacidade de um *software* ser usável (“*usability*”) está bastante relacionada com a atração sentida pela criança a partir dele.

Segundo Kishimoto (1996), Froebel introduz o brincar para educar e desenvolver a criança. “Froebel concebe o brincar como atividade livre e espontânea, responsável pelo desenvolvimento físico, moral, cognitivo, e os dons ou brinquedos, objetos que subsidiam atividades infantis. Entende, também, que a criança necessita de orientação para seu desenvolvimento”. Entretanto, coloca a educadora, “a aquisição do conhecimento, requer a auto-atividade, capaz de gerar auto-determinação que se processa especialmente pelo brincar”.

Segundo Weston (2000), com jogos, você pode ensinar conceitos complicados, princípios filosóficos e emoções que são difíceis de descrever em palavras. Os teóricos constataram ser o jogo um método para construir as capacidades do caráter, e nos listam capacidades essenciais do caráter: autoconhecimento, auto-estima, motivação interior, autoconsciência, harmonia social, sensibilidade, capacidade de tomar decisões conscientemente, de pensar de forma independente, ter alegria, capacidade de lidar com o estresse, engenhosidade e humanidade.

Guzdial; Soloway (2002) constatam um obstáculo a ser vencido pelos *softwares* educacionais, particularmente no ensino de ciência da computação: atualmente os estudantes, também chamados de geração Nintendo ou geração MTV, por serem profundamente influenciados por este tipo de fonte de tecnologia, esperam produzir mídia com gráficos, sons, animações com a qualidade que estão habituados. A dificuldade, segundo os pesquisadores, é a falta de boas plataformas de programação com multimídia, prejudicando o engajamento dos estudantes, fato esse bastante crítico para o aprendizado.

Os jogos de computador ou de video game atuais são grandes sucessos de venda entre crianças, adolescentes e adultos. Gerou-se uma indústria de entretenimento de jogos interativos de bilhões de dólares, onde as tecnologias mais

avançadas são empregadas para satisfazer este mercado, também gerando uma demanda por novas pesquisas para deixá-los cada vez mais atrativos, desafiadores e com mais realismo. O jogo Quake, de acordo com Hawkins; Astle (2001), mudou a arte de criação de jogos ao utilizar entidades totalmente tridimensionais, que podiam se mover em um mundo poligonal totalmente 3D com seis graus de liberdade. Tecnologias de síntese de imagens 3D como traçado de raios (*ray-casting*) e geração de polígonos (*polygon rendering*) são empregadas com desempenho cada vez melhor. Os conceitos e tecnologias de Realidade Virtual e Inteligência Artificial vêm sendo empregados nos jogos para aumentar a imersão e o realismo.

Os jogos atuais de computador são muito atrativos por vários fatores como o desafio ao jogador para atingir objetivos, seu realismo, grau de interatividade, disponibilidade de diferentes níveis de aptidão e a possibilidade de vários jogadores atuarem ao mesmo tempo, geralmente como competidores mas, por vezes, também como colaboradores. Neste sentido, o jogo de computador começou a ser visto como uma outra alternativa para o desenvolvimento de *softwares* educativos. Um exemplo é o Edugraph (BATTAIOLA, 2002): “*software* de ensino de conceitos de computação gráfica, com base em critérios de interfaces de jogos de computador”. Segundo Battaiola (2002), critérios de jogos de computador também podem ser considerados para *softwares* educacionais: “O desafio proporcionado pela trama, a compatibilização da interface com a trama, a sofisticação das cenas, a performance e a facilidade de interação”.

Um novo termo tem sido utilizado para descrever *softwares* nesta área de entretenimento educativo: *edutainment*.

Papert (1998), acrescenta um outro fator importante e surpreendente: os jogos são divertidos por que não são fáceis. Ele afirma que a aprendizagem é essencialmente difícil, e que ela acontece melhor quando se está profundamente engajado em atividades complexas e provocativas. Para a maioria das crianças, segundo o teórico, a razão de não gostarem das escolas não é porque o trabalho é bastante difícil, mas porque é chato. As crianças preferem coisas difíceis desde de que também sejam interessantes, ele afirma.

Outra observação importante de Papert (1998) é o benefício para o processo de aprendizagem, que se pode alcançar ao se propor para as crianças atividades em

que elas mesmas se tornem as projetistas de jogos. Segundo o mesmo, o entusiasmo aumenta quando elas próprias fazem os jogos, gerando pensamentos mais sofisticados sobre todos os aspectos envolvidos em um jogo.

2.2 Ambientes Virtuais Imersivos

A Realidade Virtual fornece interfaces mais avançadas do usuário com um mundo virtual que simula o real. Na prática, segundo Netto; Machado; Oliveira (2002), a Realidade Virtual permite que o usuário navegue e observe um mundo tridimensional, em tempo real e com seis graus de liberdade: para frente / para trás, acima / abaixo, esquerda / direita, inclinação para cima / para baixo, angulação à esquerda / à direita e rotação à esquerda / à direita.

Pinho (1999) ressalta a característica dinâmica do ambiente virtual: “os cenários se modificam em tempo real à medida que os usuários vão interagindo como o ambiente”.

A interatividade com o ambiente virtual é um ponto importantíssimo na Realidade Virtual e diversos dispositivos especiais podem ser utilizados, gerando um maior ou menor grau de imersão, como dispositivos de visualização colocados na cabeça (HMD – *head-mounted display* e HCD – *head-coupled display*), projetores, telas, luvas de dados (*data gloves*), luvas de contato (*pinch gloves*) *joysticks* 3D, sensores e dispositivos para rastreamento da trajetória e movimentos do usuário, sensores de voz para detecção de comandos de voz, óculos estereoscópicos, dispositivos *haptics* para detecção de tato onde se pode sentir pressão, textura e forma, e detectar a posição. A visualização tridimensional em um monitor de vídeo é uma forma de Realidade Virtual não imersiva que também pode ser utilizada.

Pinho (1996) coloca que a Realidade Virtual nos permite aprender visitando lugares onde jamais estaremos na vida real, talvez porque o lugar seja muito pequeno para ser visto ou muito grande para ser examinado como um todo, ou muito caro ou muito distante. Ela permite ainda que visitemos lugares em períodos diferentes de tempo.

A Realidade Virtual tem sido utilizada em aplicações de engenharia, medicina, previsão meteorológica, simulações como simulação de vôo, entretenimento, treinamento, visualização de dados, prototipagem virtual, e outras,

sendo que a cada momento surgem novos tipos de aplicação onde a Realidade Virtual é empregada.

2.2.1 Realidade Virtual Aplicada à Educação

Segundo Pinho (1996), a Realidade Virtual aplicada à educação está no início de sua caminhada e pesquisas estão sendo feitas na direção de como isto pode ser feito e de como avaliar sua contribuição para o processo de aprendizagem. No geral, comenta o teórico, parece existir um consenso de que a Realidade Virtual pode ajudar no processo de ensino.

Winn (1997) diz que não há dúvida que estudantes podem aprender em ambientes virtuais. A questão é se a Realidade Virtual imersiva e tecnologias de ponta podem ajudar estudantes a aprender matérias complexas mais facilmente, com maior sucesso e possivelmente antes do que podem com outras tecnologias. Outras questões importantes, segundo o teórico, são se a Realidade Virtual pode ajudar os estudantes a reter o que aprenderam por maior tempo e se ambientes virtuais podem compensar alguma falta de habilidade com sistemas de símbolos utilizados em classes mais tradicionais. O teórico sugere que sejam pesquisados alguns atributos da Realidade Virtual, como autonomia, sentido de presença participativa e interação, para verificar a contribuição no processo de aprendizado assim como o emprego das estratégias de aprendizagem construtivista e tradicional em ambientes virtuais.

Roussos et al. (1999) ressaltam não existir até o momento evidência clara de que a Realidade Virtual traga valor agregado ao aprendizado da criança, mas mesmo que houvesse, o custo proibitivo das tecnologias e concomitante pessoal técnico para desenvolvimento, operação e manutenção, não encontrariam lugar nos orçamentos das escolas, dominados por custos de recursos humanos.

Pinho (1996) relata algumas experiências de Realidade Virtual aplicada à educação como o projeto VRVV (Virtual Reality Roving Vehicle) do laboratório Human Interface Technology Laboratory (HITL) da Universidade de Washington onde a idéia era levar uma caminhonete com equipamentos de Realidade Virtual até 50 escolas. Neste projeto, eram dois os objetivos centrais segundo o pesquisador: prover informações sobre Realidade Virtual para crianças dando oportunidade a elas de experimentar um mundo virtual e a construção de mundos virtuais. Outros

assuntos abordados em projetos desta área relatados pelo pesquisador foram: sala de aula virtual da Universidade de Washington em convênio com o hospital de crianças de Seattle, livro virtual (Haywood Community College), ensinos de química (projeto Chemistry World da Universidade de Washington), física (projeto ScienceSpace do Virtual Environment Technology Laboratory – VETL, uma pesquisa em conjunto da Universidade de Houston, John Space Center da NASA - *National Aeronautics and Space Administration* - e Universidade George Mason), anatomia (projeto Visible Human – National Institutes of Health) e arqueologia (“Centro de Estudos de Arquitetura em Bryn Mawr”).

Segundo Allison; Hodges (2000), o impacto que a Realidade Virtual terá na educação pública ainda não é claro. Os pesquisadores descrevem o projeto Virtual Gorilla onde conceitos sobre o comportamento e interações sociais dos gorilas são ensinados. Outros projetos educacionais em ambientes de Realidade Virtual imersivos relatados são sobre física, álgebra espacial (projeto Spatial Algebra do laboratório HITL da Universidade de Washington), jardinagem (projeto NICE – Narrative-based, Immersive, Constructionist / Collaborative Environments do laboratório Electronic Visualization Lab - EVL - da Universidade de Illinois), formato do planeta Terra (também do laboratório EVL) além do já mencionado projeto ScienceSpace. Os pesquisadores destacam um problema para se aplicar Realidade Virtual imersiva em educação: leva-se muito tempo e esforço para gerar conteúdo suficiente para fazer um sistema que ensine conceitos valiosos.

Jackson; Fagan (2000) apresentam o projeto Global Change World (GCW) do laboratório HITL da Universidade de Washington, um ambiente virtual imersivo de aprendizagem projetado para fornecer interação simultânea multiusuário. O GCW, segundo os pesquisadores, fornece uma experiência colaborativa empregando capacetes visuais e sistema de comunicação por voz, para professores e alunos comunicarem-se em um ambiente virtual tridimensional.

O projeto NICE (ROUSSOS et al., 1999); (JOHNSON et al., 1998) procura criar um ambiente de aprendizagem baseado nas teorias educativas de construtivismo e aprendizagem colaborativa disponibilizando cenários de Realidade Virtual onde crianças constroem e cultivam ecossistemas. O objetivo do projeto é construir um ambiente de teste para exploração da Realidade Virtual como uma mídia de

aprendizagem de educação básica. O projeto para crianças de seis a dez anos foi realizado em ambiente imersivo de múltiplas projeções, onde cinco ou seis crianças podiam participar das atividades.

Segundo os pesquisadores, NICE é flexível o suficiente para ser eventualmente expandido em um sistema de autoria, mas para ser efetivo, ele necessitaria ser usado por um número menor de aprendizes por um tempo maior.

As crianças no projeto NICE utilizam óculos obturadores e um *joystick* simulando uma varinha com três botões, um para pegar e plantar, um para alterar tamanho e um para saltar. No jardim existem cestos com sementes para as crianças as escolherem, levar e deixar cair no solo. O vegetal correspondente começa a crescer e as crianças devem tomar cuidado para que as plantas não fiquem muito próximas uma das outras e que recebam água e luz do sol. As crianças podem mover nuvens para regar as plantas, utilizar o sol para fornecer luz e limpar ervas daninhas. O jardim é persistente e o seu tempo de crescimento é configurável.

NICE suporta colaboração distribuída em tempo real. Crianças em localizações remotas podem interagir com o jardim. A presença de usuários remotos é estabelecida usando um avatar (representação gráfica do corpo de uma pessoa no mundo virtual). Adicionalmente, comunicação por voz é habilitada por conexão de áudio em tempo real.

As crianças aprenderam a utilizar a varinha num intervalo de tempo de 2 a 7 minutos, sendo que algumas não gostaram “da coisa que tinha de se mover com”, e dos “óculos que caiam”. O líder dos grupos, escolhido aleatoriamente, era responsável por manejar a varinha e navegar, sendo permitido um tempo de 10 minutos de prática. As crianças, que executavam a função de líder, se engajavam mais enquanto as demais ficavam mais distraídas. Cada sessão durava em média 30 minutos sendo que um professor utilizava um avatar compartilhando o mesmo espaço virtual remotamente, com a função de controlar o tempo, guiar os grupos e manter as crianças focadas na tarefa de plantar. A comunicação com o usuário remoto era estabelecida por meio de microfones escondidos.

Em relação à colaboração, estudos com pares de crianças apresentaram excelentes níveis de colaboração ao contrário do comportamento das classes. O

avatar do professor se mostrou útil principalmente para dar dicas e manter as crianças focadas.

Segundo os pesquisadores do projeto NICE, os pesquisadores de aprendizagem em ambientes virtuais se deparam com um desafio difícil. Por um lado existe forte necessidade de se demonstrar “valor agregado” ao aprendizado associado com tecnologias de Realidade Virtual, por outro lado existem poucas razões para fazer a tecnologia de Realidade Virtual carregar objetivos já alcançados pela pedagogia convencional. Os pesquisadores sugerem para as pesquisas nesta área, atenção sobre quatro aspectos: “o objetivo de aprendizagem deve ser importante”; “o objetivo de aprendizagem deve ser difícil”; “o objetivo de aprendizagem deve ser notadamente aumentado com a introdução de Realidade Virtual imersiva” e “ambientes de aprendizagem baseados em Realidade Virtual devem ser informados por pesquisas contemporâneas das ciências de aprendizagem, práticas contemporâneas em educação, e por realidades práticas de organização da escola e seus custos.

2.2.2 Interação

A interação com um mundo virtual em ambiente imersivo é um fator crítico para o sucesso das aplicações. Embora o crescimento nesta área seja sensível nos últimos anos, ainda há muito o que avançar nas técnicas de interação para que o usuário interaja com naturalidade e eficiência.

Segundo Pinho (2000), as técnicas de interação em ambiente imersivo devem permitir ao usuário, executar ações de seleção, manipulação e navegação, preferencialmente aproveitando seus conhecimentos sobre o mundo real. *Seleção* é o processo de se identificar um objeto ou conjunto de objetos sobre o qual deseja-se operar. *Manipulação* é uma tarefa que acarreta mudança nas propriedades geométricas de um ou mais objetos, enquanto a *navegação* é o processo de se mudar o ponto de vista do observador.

A dificuldade, ressalta Pinho (2000), é a total falta de restrições nos ambientes imersivos, e a incapacidade inerente aos dispositivos de Realidade Virtual de captar todas as informações possíveis de serem produzidas pelo usuário. O pesquisador exemplifica algumas tarefas de interação realizadas com naturalidade no

ambiente real, mas que apresentam dificuldades no ambiente virtual: “a seleção de ferramentas é complicada”, “a tecnologia de reconhecimento de voz ainda é precária”, “a entrada de textos e números é pouco desenvolvida”.

Mine apud Pinho (2000) define três categorias de tarefas interativas:

- a) Interação direta: o usuário utiliza o corpo (mãos, braços, cabeça, etc.), sendo necessário suporte ao rastreamento das ações do usuário.
- b) Interação com controles físicos: o usuário utiliza botões, *joysticks*, pedais, etc.
- c) Interação com controles virtuais: o usuário utiliza dispositivos físicos representados visualmente.

Pinho et al. (1999), propõe “mapear movimentos de um usuário diretamente para movimentos dentro do mundo virtual, sem que este precise memorizar metáforas e seqüências de botões para interagir no ambiente”. Nesta pesquisa foi implementado um protótipo de uma bicicleta virtual construída a partir de uma bicicleta real presa a um tripé, gerando-se um dispositivo de navegação baseado na leitura dos movimentos do usuário na bicicleta e exibição de imagens usando óculos de Realidade Virtual.

Outro exemplo de navegação em ambiente virtual onde o usuário caminha fisicamente é o GSS (*Ground Surface Simulator*) proposto por NOMA; SUGIHARA; MIYASATO (2000). O GSS é um dispositivo de locomoção virtual que pode melhorar comunicações interpessoais remotas através de um novo método de comunicação chamado de Tel-E-Merge. Este método permite a conversação entre pessoas remotamente separadas enquanto elas caminham juntas, focando-se no senso de locomoção que é sentido quando se caminha em solo real.

A seleção de objetos em ambiente virtual imersivo, segundo Pinho (2000), ocorre em duas etapas: a indicação do objeto pelo usuário e a confirmação da seleção. A indicação, segundo o pesquisador, pode ser feita de várias maneiras, como a técnica de *ray casting* onde o objeto é apontado por uma espécie de raio de *laser* que sai da mão do usuário; pode ser feita tocando-se objetos próximos com o usuário segurando um sensor de posição em sua mão. Outras opções são comandos de voz ou seleção por lista de objetos através de menus. A confirmação da seleção, esclarece o pesquisador, pode ser feita com gestos, botões, ou até mesmo com um dispositivo de tempo que confirma a seleção após determinado tempo em que um objeto fica sendo

apontado. De qualquer modo, ressalta o pesquisador, nas duas etapas é necessário um retorno ao usuário (*feedback*), o que pode ser feito alterando-se a cor do objeto selecionado, com moldura ou marcadores ao seu redor. Um sinal sonoro também pode ser utilizado na etapa de confirmação.

A manipulação de objetos, segundo Pinho (2000), pode ser feita por manipulação direta, “pegando-se” um objeto próximo, movendo-o e largando-o na posição desejada. Outras sugestões do pesquisador são manipulação por raios ou através de dispositivos físicos como o *mouse* tridimensional ou rastreador.

Dickerson et al. (2002) do Virtual Reality Application Center (VRAC) da Universidade do Estado de Iowa, pesquisam dispositivos sem fio de pequena potência para interação em ambientes virtuais imersivos. Dispositivos de rastreamento, computadores palm-top, óculos infra-vermelho obturadores, varinha de seleção (6-D *wand*), microfone e luva de dados são projetados de modo a não interferir uns nos outros e com latência* não perceptível pelo homem.

2.2.3 Rastreamento

O rastreamento ou *tracking* é um componente fundamental em um ambiente imersivo de Realidade Virtual, onde a interação do usuário com o mundo virtual deve ser a mais intuitiva possível. É o rastreamento dos movimentos do usuário que permite a este interagir com naturalidade, contribuindo com o grau de imersão oferecido ao usuário.

A escolha do sistema de rastreamento é uma questão que deve ser tomada cuidadosamente, segundo Cruz-Neira (1997), pois ele é a maior fonte de erros e atrasos em uma experiência virtual, afetando sua performance e gerando desconforto.

O usuário deve ter liberdade para navegar ou movimentar naturalmente e é neste ponto que o sistema de rastreamento se torna necessário, captando os movimentos com a performance suficiente para que o sistema calcule, em tempo real, as projeções do mundo virtual de acordo com os novos pontos de vista do usuário. Os rastreamentos mais comuns são os da cabeça e das mãos.

* Latência é o tempo de atraso total entre a ação do usuário e a resposta do sistema.

O rastreamento, como visto no item sobre interação em ambiente imersivo, também pode ser utilizado nas interações de seleção e manipulação de objetos em ambiente imersivo.

Os sistemas de rastreamento, descrevem Netto; Machado; Oliveira (2002), utilizam sensores colocados sobre partes do corpo ou objetos, que emitem sinais a uma caixa controladora que processa estes sinais e os envia ao computador. Estes sistemas são classificados de acordo com a tecnologia que empregam. As mais comuns são técnicas eletromagnéticas, mecânicas, acústicas, ópticas e processamento de imagens. .

2.2.4 Estereoscopia

Uma imagem estereoscópica é aquela em que os olhos esquerdo e direito do observador, visualizam a cena cada um com seu ponto de vista, assim como o ser humano visualiza os objetos no mundo real. O cérebro, então, sintetiza as duas imagens em uma única e com profundidade estereoscópica.

A palavra estéreo tem origem grega *stereós*, que significa sólido. Desde a Antiguidade os teóricos indagavam por que o homem não via duas imagens, mas foi Wheatstone (1838), quem explicou que o problema, na verdade era uma solução, demonstrando que o cérebro fundia as duas imagens díspares da retina, em uma única “imagem sólida”. Um exemplo de disparidade das imagens pode ser observada na figura 1.

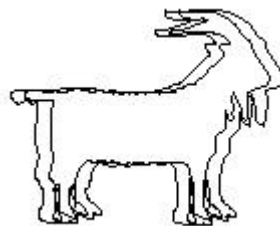


Fig. 1 – Disparidade das imagens na retina (STEREOGRAPHICS, 1997)

A visualização estereoscópica, em sistemas de Realidade Virtual, contribui para aumentar o grau de imersão.

Em imagens estereoscópicas geradas por computador, o sistema projeta a imagem do olho esquerdo e em seguida a do olho direito, numa alta frequência. Se a

freqüência for de 120 Hz, por exemplo, 60 imagens por segundo são projetadas para cada olho. Se visualizarmos o monitor, veremos as duas imagens sobrepostas. Através de óculos obturadores, a imagem projetada sob o ponto de vista do olho esquerdo é visualizada apenas pelo olho esquerdo, enquanto que a imagem projetada sob o ponto de vista do olho direito, é visualizada apenas pelo olho direito. Então, o cérebro humano junta as duas imagens e o efeito estereoscópico é percebido. Na figura 2, um emissor infravermelho do dispositivo CrystalEyes da StereoGraphics Corporation, envia sinais de sincronismo para os óculos obturadores impedirem a visão do olho esquerdo quando a imagem projetada no monitor de vídeo é a do olho direito, e vice-versa, isto na freqüência vertical de 120 Hz.

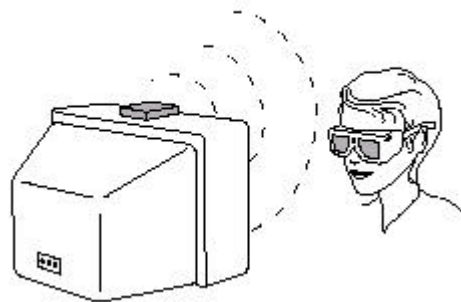


Fig. 2 – Imagens estereoscópicas por computador (STEREOGRAPHICS, 1997)

A paralaxe, distância entre as imagens do olho esquerdo e direito, é que causa o efeito estereoscópico. Se a paralaxe for zero, então percebemos o objeto na tela. Se a paralaxe for negativa, então percebemos o objeto antes da tela, mas se a paralaxe for positiva, percebemos o objeto atrás da tela. A figura 3 apresenta os três tipos de paralaxe.

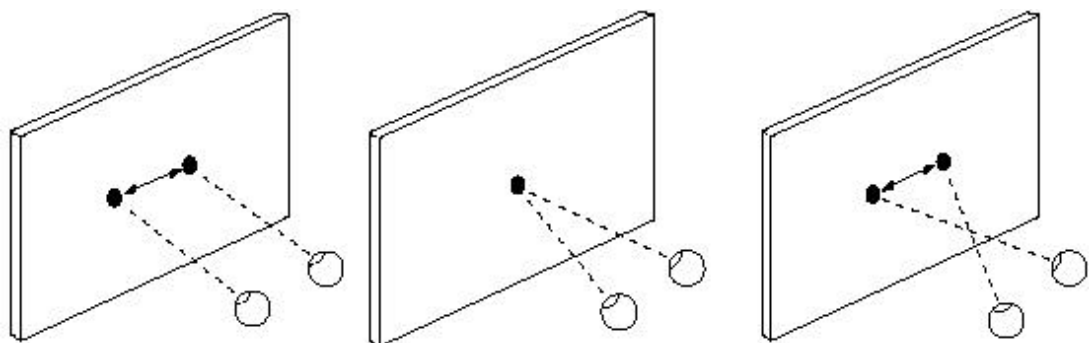


Fig. 3 – Paralaxes positiva, zero e negativa (STEREOGRAPHICS, 1997)

As imagens estereoscópicas podem causar desconforto ao usuário. A prática mostra que se deve usar o menor paralaxe possível, mantendo o efeito de profundidade desejado. O objeto central da cena deve estar a maior parte do tempo no centro da tela, ou seja, com paralaxe zero. Isto para não forçar os olhos do usuário em demasia.

Outro fator importante para gerar uma visão agradável ao usuário, é permitir que ele regule a distância entre as câmeras que projetam as imagens do olho direito e esquerdo.

2.2.4 Som tridimensional

Em Realidade Virtual, o sistema de áudio contribui com o grau de imersão do usuário através de som tridimensional. Segundo Naef; Stadt; Gross (2002), o som espacial fornece uma importante dica ao ser humano, aumentando o sentido de presença em ambientes virtuais.

Os pesquisadores listam alguns pré-requisitos de um sistema de áudio adequado para ambientes imersivos de múltiplas projeções: a localização precisa da fonte de som no espaço 3D, simulação acústica do ambiente virtual, entrada de som ao vivo para aplicações colaborativas mais velocidade e eficiência para permitir um número razoável de fontes de som simultaneamente. Cruz-Neira; Sandin; Defanti (1993) alertam que na prática a localização do som neste tipo de ambiente é comprometida por reflexões que saem das paredes de projeção.

Sons podem ser ligados a objetos de maneira que o usuário escute um som vindo da direção de um objeto e com intensidade proporcional à proximidade do usuário em relação ao objeto.

Segundo Hawkins; Astle (2001), a determinação da posição de um som pelo ser humano depende de vários fatores, incluindo a visão. Alguns fatores vêm do próprio som:

- a) À medida que a fonte sonora se move para longe do ouvinte, o volume do som percebido diminui a uma taxa fixa.

- b) Um som que vem de uma fonte à direita do ouvinte, soa mais alto no ouvido direito do que no esquerdo, assim como é percebido um pouco antes pelo ouvido direito do que pelo esquerdo.
- c) A forma do ouvido garante que sons vindos de trás do ouvinte são levemente abafados em comparação com sons vindos da sua frente.

Na figura 4, observa-se duas maneiras diferentes de se configurar os alto-falantes em ambiente de múltiplas projeções, visando-se implementar som tridimensional.

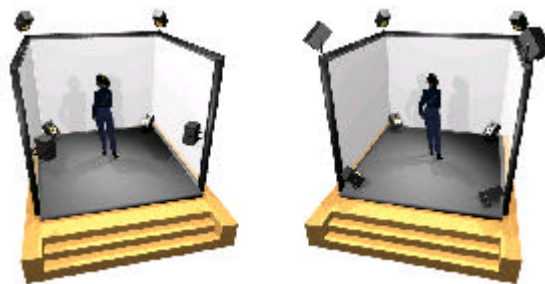


Fig. 4 – Ambiente imersivo com alto-falantes (NAEF; STAADT; GROSS 2002)

2.2.5 Animação

A técnica de animação consiste em uma série de imagens que, sendo visualizadas sequencialmente pelo homem, geram o efeito de movimento. Cada imagem é chamada de quadro ou *frame*. Quando se assiste a uma fita de vídeo gravada no padrão NTSC (*National Television Standards Committee*), a cada segundo são apresentados trinta quadros, ou seja, trinta *frames per second* (quadros por segundo).

Para se criar uma animação, muitos quadros são necessários. Para reduzir esta carga de trabalho do animador, surgiu a técnica de *keyframing*. Os estúdios de animação perceberam que seriam mais produtivos se o animador desenhasse apenas os quadros principais, ou chaves, enquanto assistentes se encarregavam dos quadros intermediários, ou *inbetween frames*.

Os pesquisadores de animação por computador, segundo Lasseter (1987), desenvolveram técnicas de animação 2D a partir da animação tradicional. Técnicas como *storyboarding*, *keyframe*, *inbetweening*, foram aplicadas no computador. Como a animação 3D usa modelos 3D em vez de desenhos 2D, poucas técnicas da computação tradicional foram aplicadas. Segundo o pesquisador, os primeiros

sistemas de animação foram baseados em *scripts*. Posteriormente surgiram sistemas de animação *keyframe* permitindo às pessoas produzir animações por computador com muito mais qualidade. Porém, o pesquisador diz que muitas animações ruins são produzidas no computador por causa da não familiaridade com princípios fundamentais da animação, usados por mais de cinquenta anos em desenhos manuais pelos animadores. Dicas de animação tradicional como, por exemplo, esticar e achatar o objeto durante a animação, são apresentadas pelo pesquisador visando a sua utilização na animação 3D por computador com a técnica *keyframe*.

Existem duas maneiras, segundo Thalmann; Thalmann (1998), de se abordar a animação por computador. Uma é a evolução da animação tradicional estendida ao computador. A outra é a simulação das leis da física, especialmente as da mecânica. Nenhuma das duas é melhor que a outra, segundo a teórica. Ela também diz que a animação por computador tem sido considerada por muitos anos uma nova mídia para propaganda e filmes de efeitos especiais. Porém, recentemente com o desenvolvimento de poderosos computadores, novas áreas como multimídia, jogos interativos e Realidade Virtual têm se preocupado com animação interativa em tempo real.

2.2.6 Múltiplas Projeções

Os sistemas de múltiplas projeções como o CAVE (CAVE Automatic Virtual Environment) (CRUZ-NEIRA et al., 1992) apresentado na figura 5, surgiram como propostas de ambientes de Realidade Virtual totalmente imersivos. Netto; Machado; Oliveira (2002) descrevem bem o que é um ambiente de múltiplas projeções: “uma sala em que paredes, teto e chão são telas semi-transparentes onde as imagens são projetadas, permitindo que uma ou mais pessoas fiquem imersas no ambiente virtual. A projeção é feita por projetores posicionados atrás das telas e pode ser estereoscópica, exigindo dos usuários o uso de óculos obturadores”.

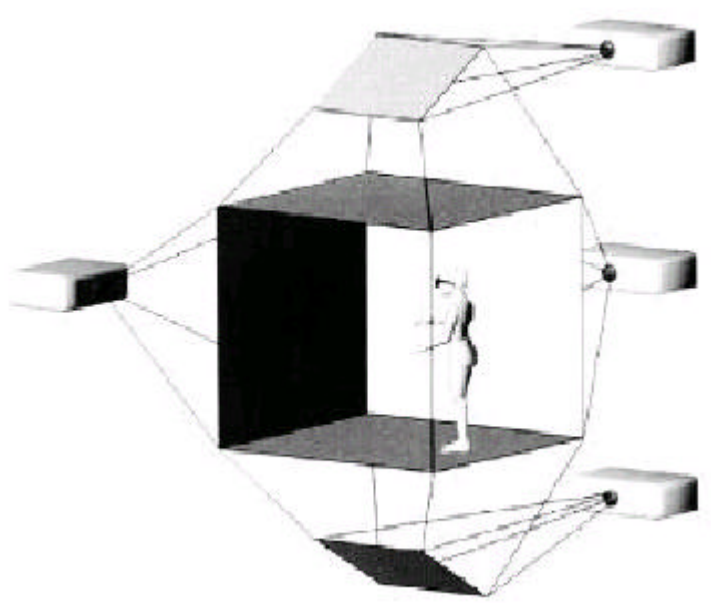


Fig. 5 – Projeções em um CAVE (CRUZ-NEIRA et al., 1992)

A primeira iniciativa semelhante na América Latina foi a CAVERNA Digital (ZUFFO et al., 2001); (SANTOS et al., 2001); (SOARES et al., 2002), um cubo de dimensões 3m x 3m x 3m com cinco projeções (4 paredes e piso), comportando até seis usuários em seu interior compartilhando a mesma experiência virtual. A figura 6 apresenta o layout da CAVERNA Digital enquanto que a figura 7 apresenta usuários interagindo com a CAVERNA Digital e seu console.

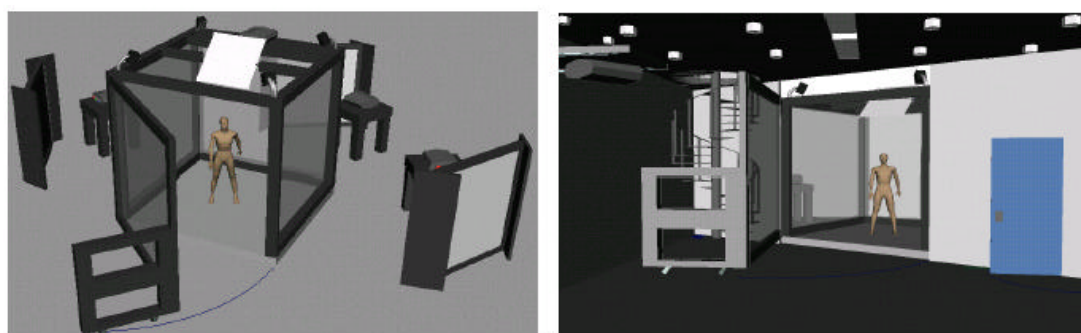


Fig. 6 – *Layout* da CAVERNA Digital (ZUFFO et al., 2001)



Fig. 7 – Usuários e operadores na CAVERNA Digital (ZUFFO et al., 2001)

Outros recursos, além da projeção de imagens sincronizadas, geralmente são fornecidos para facilitar a interação e aumentar a imersão que o usuário sente neste ambiente: síntese de imagens tridimensionais estereoscópicas, sistemas de rastreamento da localização do usuário e seus movimentos, sistemas de interação como luvas de dados e *joysticks*, e ainda, som tridimensional (3D).

Uma das grandes vantagens deste ambiente de múltiplas projeções é a possibilidade de vários usuários compartilharem a mesma experiência, como enfatizaram Cruz-Neira et al. (1992): “Um dos mais importantes aspectos da visualização é a comunicação. Para a Realidade Virtual se tornar uma ferramenta efetiva e completa, ela tem que permitir mais de um usuário no mesmo ambiente”. Outras vantagens destacadas pelos pesquisadores são as características de alta resolução, largo campo de visão, e interface não invasiva.

2.3 Trabalho Colaborativo

Segundo Bannon e Schmidt apud McManus (1997), *Computer-Supported Cooperative Work* (CSCW) ou Trabalho Cooperativo Apoiado por Computador, estuda trabalho de indivíduos em conjunto apoiados por meio de sistemas computadorizados.

Segundo Glass e Putnam apud McManus (1997), *Collaborative Learning* (CL) ou Aprendizagem Colaborativa, estuda estudantes trabalhando juntos em pequenos grupos heterogêneos visando um objetivo acadêmico comum. McManus relata que a aprendizagem cooperativa já existe há muito tempo, e que muitos pesquisadores propõem: os estudantes aprendem de maneira ótima quando interagem

com outros estudantes ou professores de maior habilidade, especialmente através da discussão de suas atividades.

Segundo O'Malley e Koschmann apud McManus (1997), *Computer Supported Collaborative Learning* (CSCL) ou Aprendizagem Colaborativa Apoiada por Computador, estuda o uso da tecnologia de computação na aprendizagem colaborativa. Segundo McManus, CSCL é uma combinação de CSCW e CL. CSCL, engloba sistemas para projeto de trabalho colaborativo, onde se permite que estudantes desenvolvam projetos compartilhados em um ambiente de rede de computadores e também reúne sistemas de criação de classes virtuais, onde locais de encontros eletrônicos são disponibilizados através de parceiros virtuais. A pesquisadora cita alguns sistemas CSCL como o CSILE (Computer Supported Intentional Learning Environments) e o CoVis (Learning through Collaborative Visualization).

Gava e Menezes (2002) propõem Moonline, um ambiente baseado na WEB para apoio à construção coletiva de conhecimento. Segundo os pesquisadores, “os ambientes cooperativos podem ser usados para mudar o cenário tradicional de ensino, que tendo o professor como ponto central e ‘detentor’ da informação, viabilizando meios alternativos de ensino-aprendizagem”. Os pesquisadores citam alguns sistemas CSCL como o CSILE, NICE, EVA (Virtual Learning Environment), COSOFT (Computer Support for Face-toFace Teaching)

O MUSEU VIRTUAL é uma ferramenta demonstradora de como a tecnologia da realidade virtual e a Internet, por meio da web, podem ser utilizadas para implementar sistemas de aprendizagem colaborativa apoiada por computador. Esta ferramenta provê uma linguagem visual básica, permitindo criar objetos e programar o comportamento complexo de objetos sem o conhecimento prévio de uma linguagem de programação (WAZLAWICK et al., 2001).

Pinho e Freitas (2001) pesquisaram viabilizar que mais de um usuário possam interagir sobre um mesmo objeto ao mesmo tempo em um ambiente imersivo. Normalmente, segundo os pesquisadores, algum mecanismo impede o acesso simultâneo ao mesmo objeto, permitindo apenas a manipulação de um usuário por vez.

Segundo Suthers (2001), pesquisas consistentes têm demonstrado o valor da colaboração na aprendizagem e várias arquiteturas síncronas e assíncronas têm sido exploradas.

Com relação ao acoplamento entre as atividades de diferentes os usuários, Suthers lista três níveis:

- a) *Rigoroso* (“*what you see is what I see* – WYSIWIS”): todos os usuários têm a mesma visão. É adequado apenas para dois ou três usuários com atividades altamente acopladas.
- b) *Relaxado*: os diferentes usuários podem ter vistas diferentes à medida em que navegam ou executam suas próprias operações como editar ou mover um objeto.
- c) *Acoplamento em nível de modelo* (“*what you model is what I model* – WYMIWIM”): os usuários visualizam o mesmo modelo semântico compartilhado, mas as visões podem ser totalmente diferentes.

O teórico descreve quatro arquiteturas classificando-as de acordo com o modelo MVC - *Model-View-Controller*. Modelo (*model*) representa o modelo semântico do problema de interesse. Visão (*view*) apresenta o modelo através de alguma representação visual. Controlador (*controller*) permite ao usuário modificar o estado corrente do modelo. A arquitetura que oferece maior flexibilidade para aplicações colaborativas de aprendizagem é a híbrida, segundo a conclusão do pesquisador.

Arquitetura Centralizada – Esta arquitetura, apresentada na figura 8, fornece apenas uma aplicação e distribui cópias da interface gráfica. O controle do modelo, o modelo e a visão permanecem todos no mesmo computador. Ela obriga o acoplamento rigoroso. Os usuários têm que compartilhar o teclado e *mouse*. Apesar de ser apropriada para treinamento um-a-um, ela não é aconselhada para aplicações colaborativas de aprendizagem por não dar a liberdade ao usuário de trocar livremente de modo de trabalho paralelo e trabalho em grupo.

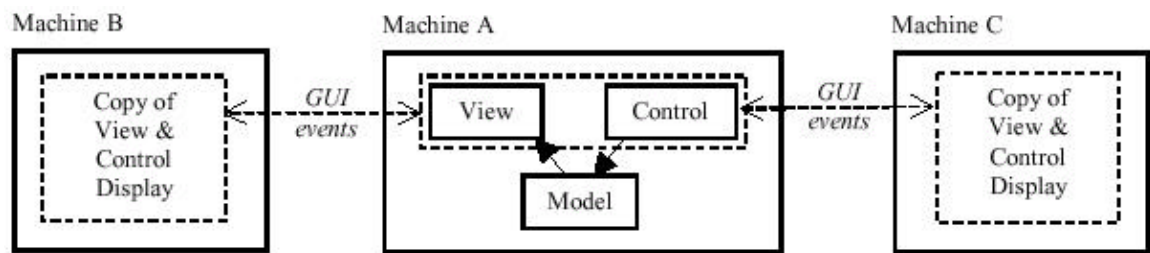


Fig. 8 – Arquitetura Centralizada (SUTHERS, 2001)

Arquitetura Replicada – Nesta arquitetura, apresentada na figura 9, a aplicação é instalada em cada estação e algum mecanismo de sincronização é fornecido. Ela é melhor adequada ao acoplamento rigoroso do que ao relaxado, e melhora o aproveitamento dos recursos de rede. Permite que o sincronismo seja ligado e desligado conforme a vontade do usuário, que pode trabalhar sem a rede (*standalone*).

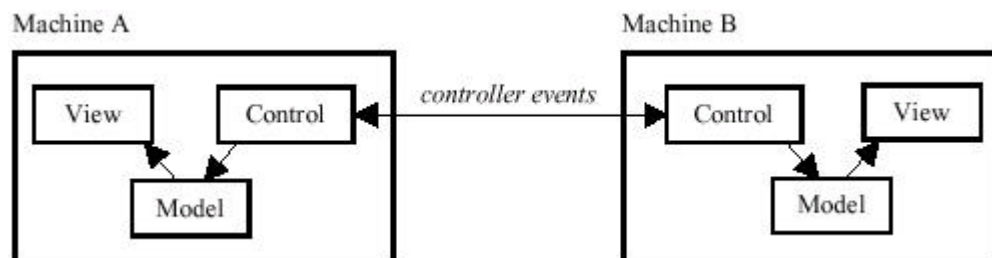


Fig. 9 – Arquitetura Replicada (SUTHERS, 2001)

Arquitetura Distribuída – Nesta arquitetura, apresentada na figura 10, os componentes de modelo, controle do modelo e visão são distribuídos em múltiplos servidores. Tipicamente o modelo é compartilhado em um servidor. Sendo o acoplamento em nível de modelo ele suporta WYMIWIM onde os usuários colaboram em um mesmo modelo. A desvantagem é necessidade de uma rede confiável, não permitindo o trabalho em um computador único (*standalone*).

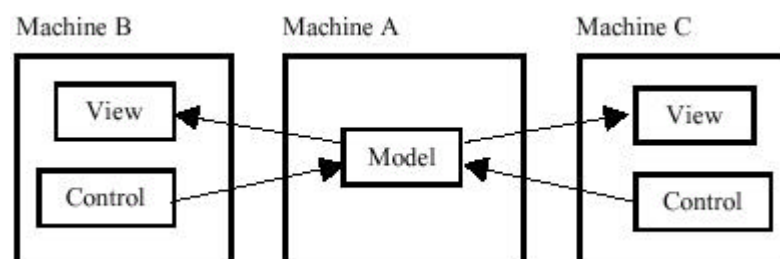


Fig. 10 – Arquitetura Distribuída (SUTHERS, 2001)

Arquitetura Híbrida – Arquitetura híbrida entre a replicada e a distribuída. A sincronização é em nível de modelo permitindo o trabalho em um único computador (*standalone*). Esta arquitetura é apresentada na figura 11.

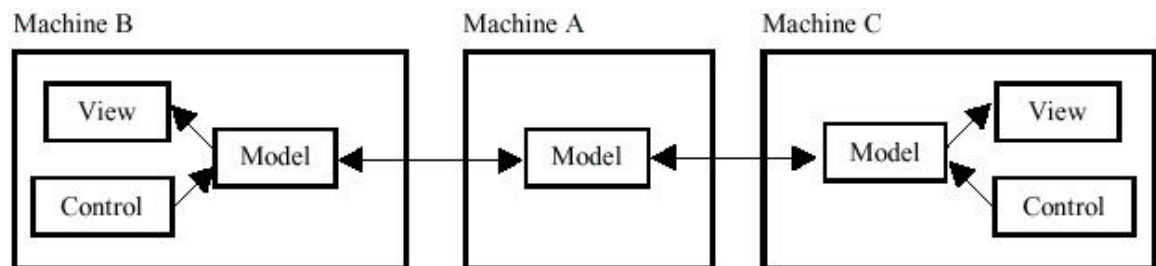


Fig. 11 – Arquitetura Híbrida (SUTHERS, 2001)

2.4 Linguagens de Programação e Programas Tradutores

Seria muito amigável se o homem pudesse comunicar-se com o computador através da linguagem natural em que ele está acostumado a falar, porém, segundo o teórico José Neto (1987), os programas seriam de custo proibitivo se toda a generalidade das linguagens naturais se mantivesse nas linguagens de programação, isto apesar dos avanços tecnológicos nesta área, encabeçados pelas pesquisas na área de inteligência artificial.

É fundamental que uma nova linguagem de programação seja descrita de forma completa, através de uma notação formal, não permitindo ambigüidades.

Um programa, escrito em uma determinada linguagem de programação, deve ser traduzido por um programa para permitir a sua execução. O programa que realiza esta tradução é chamado de compilador ou interpretador, sendo que a principal diferença entre um compilador e um interpretador, é que o primeiro gera um código executável em determinada plataforma de computação e o segundo executa diretamente o programa em um ambiente de execução.

José Neto (1987) apresenta a terminologia para o estudo de linguagens de programação: *‘Linguagem é uma coleção de cadeias de símbolos, de comprimento finito. Estas cadeias são denominadas sentenças da linguagem, e são formadas pela justaposição de elementos individuais, os símbolos ou átomos da linguagem’*. Ao conjunto de leis de formação das cadeias dá-se o nome de *gramática*, e a regra de *aceitação* de cadeias dá-se o nome de *reconhecedor*.

Para a construção de um compilador ou interpretador, segundo o mesmo teórico, recomenda-se que três funções básicas sejam implementadas. A análise léxica, responsável pela interface com o texto-fonte, extraíndo e classificando os átomos da gramática. A análise sintática, geralmente o coração do compilador, responsável pela análise da seqüência de átomos, identificando as sentenças de acordo com a gramática da linguagem. A terceira função é a análise semântica, onde o significado das sentenças do programa original é implementado em um ambiente de execução, no caso de um interpretador; ou é traduzido em um código executável, no caso do compilador.

2.5 Exemplos de Ambientes Virtuais de Autoria

Esta pesquisa visa conhecer algumas das ferramentas de informática disponíveis para uso no ensino fundamental que possam trazer subsídios para a concepção de um novo sistema.

2.5.1 SuperLogo

Para conhecer o ambiente Logo, utilizou-se a versão 6.4h do Microsoft® Windows Logo da empresa Softronix Inc. Encontrou-se também a versão *SuperLogo* 3.0 disponível para *download*, em português traduzida e adaptada pelo NIED da Universidade Estadual de Campinas, a partir da versão 6.4d da Softronix. O núcleo desta versão foi desenvolvido por Brian Harvey da Universidade da Califórnia. NIED (1990) apresenta a linguagem Logo, conceitos, comandos básicos e exemplos.

A linguagem Logo, nesta versão, é interpretada fornecendo uma série de comandos, muitos deles gráficos, para a sua tartaruga que tem o papel de um cursor, como por exemplo *parafrente (pf)*, *paratrás (pt)*, *paradireita (pd)*, *paraesquerda (pe)*, *useborracha (ub)*, *desapareçatat (dt)*, *tartaruga (tat)*. A figura 12 apresenta a tela do SuperLogo com alguns comandos.

Um aspecto interessante na filosofia Logo, contado por NIED (1983), é que a criança aprende ensinando a tartaruga a resolver algum problema. Desta maneira se dá liberdade à criança e incentiva-se a mesma a aprofundar-se no conteúdo.

A linguagem é bastante simples, permitindo que a criança rapidamente conheça seus principais comandos, interativamente experimentando-os e recebendo suas respostas de imediato.

A tartaruga basicamente faz desenhos 2D na tela do computador, o que é bastante familiar e amigável para uma criança. Associações com o uso de lápis e borracha, por exemplo, facilitam ainda mais o uso do Logo.

Os comandos de repetição e a criação de procedimentos aumentam o poder da linguagem para a resolução de tarefas mais complexas, assim como outros recursos de linguagens de programação auxiliam, como variáveis e comando condicional.

Porém, esta versão do Logo não se restringe a duas dimensões, pois com o comando *Logo3D* entra-se no modo de perspectiva, habilitando a criação de desenhos em 3D. Três tartarugas invisíveis, representando os olhos do usuário, o alvo do olhar do usuário e a posição da luz, são criadas. Através de comandos em 3D, como *mudexyz* e *mudedireção*, o autor pode desenhar um cubo por exemplo.

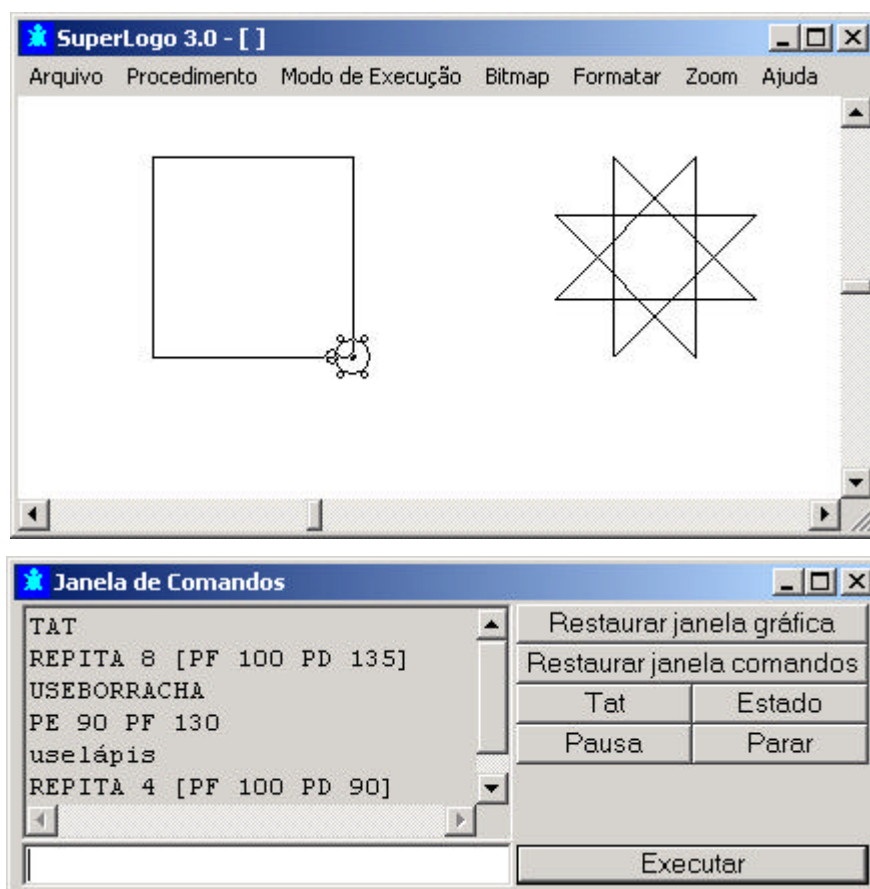


Fig. 12 – Exemplo de comandos no ambiente SuperLogo

2.5.2 Imagine

Imagine é uma nova geração do ambiente de linguagem Logo. Ele fornece uma grande variedade de atividades como: pintura, animação, programação com a linguagem Logo, criação de aplicações multimídia, comandos de voz, tanto de entrada quanto de saída, modelagem e outros. O objetivo principal é fornecer a estudantes, professores e desenvolvedores uma ferramenta geral para aprendizagem.

A linguagem Logo nesta versão foi acrescida de estruturas orientadas a objetos. O mundo consiste de objetos que respondem a mensagens a eles enviadas. Criam-se os objetos a partir de classes primitivas pré-definidas ou novas classes derivadas das primitivas, através do comando *newClass*.

O ambiente também fornece eventos pré-definidos percebidos por determinadas classes, como *OnClick*, *onCollide*, *onVoiceCommand*, que são eventos disparados por um clique do *mouse*, por colisão entre duas tartarugas e, recepção de um comando de voz.

Recursos de multimídia para tocar vídeo, som ou música são fornecidos. A figura 13 apresenta a tela do ambiente *Imagine*.

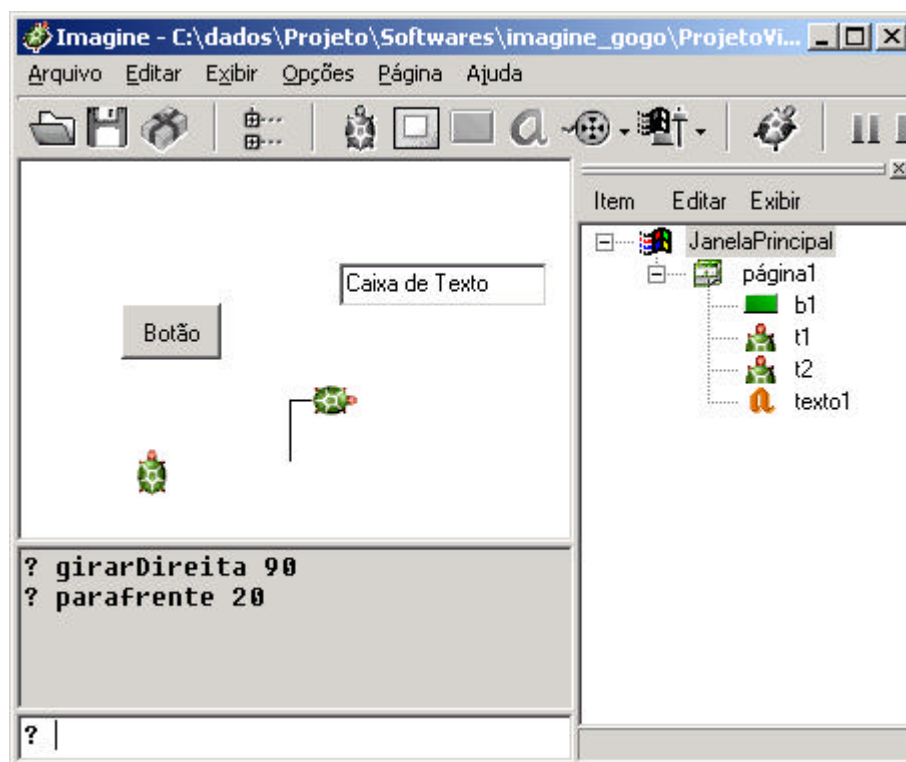


Fig. 13 – Exemplo de projeto no ambiente Imagine

2.5.3 Alice

Alice (2003) é um ambiente gráfico tridimensional programável e interativo, desenhado com o objetivo de tornar fácil para iniciantes o desenvolvimento de mundos virtuais em três dimensões. A versão utilizada neste trabalho é a Alice 99.

Ele foi construído pelo grupo de pesquisa Stage 3 (2003) na Universidade de Carnegie Mellon, porém foi originalmente desenvolvido pela Universidade de Virginia. Admirável é a missão do grupo multidisciplinar Stage 3, que se propõe a explorar e desenvolver mecanismos para aproximar a tecnologia ao homem, tornando-a mais efetiva e prazerosa, dentro do limite de se trabalhar com diversão:

“Our mission:

To have fun. Within that constraint, to explore and develop the mechanisms by which humans can more effectively and enjoyably interact with technology” (STAGE 3, 2003)

Como Conway; Pausch (1997) dizem, programas gráficos 3D interativos são difíceis de aprender e difíceis de serem alterados. Para endereçar este problema Alice abstrai as complexidades das Interfaces de Programa de Aplicação (API) gráficas através de uma arquitetura orientada a objetos e fornece recursos para imediata alteração de uma cena 3D. Realmente, o usuário rapidamente entende os conceitos de hierarquia de objetos e operações que lhes podem ser aplicadas. Também é com satisfação que ele vê um objeto sendo alterado sob seu comando, como, por exemplo, na operação *move*. Cuidados foram tomados para que o ambiente virtual se aproxime do ambiente real, para não causar estranheza ao usuário, como por exemplo, ao inserir um novo objeto na cena, este é colocado no chão, evitando-se deixar um objeto voando no céu.

As coordenadas X, Y e Z não estão presentes na API do Alice sendo que direções são fornecidas por palavras reservadas *forward, back, left, right, up, down*, lembram os comandos do programa Logo de Papert.

O público alvo deste projeto são estudantes de dezenove anos de idade de cursos de graduação, com exceção de engenharia, com nenhuma ou pouca experiência em programação (CONWAY, 1997). Porém, Alice tem feito sucesso

com as crianças. A sua tela de abertura composta por um terreno (plano com textura) e um céu azul (fundo azul) não intimidam o usuário, e seu tutorial com personagens amigáveis 3D, mostra os primeiros passos de como trabalhar com a ferramenta. O que mais impressiona neste programa é seu apelo amigável, cheio de estímulos para crianças e adultos. É exemplo de ferramenta que alia o lúdico ao pedagógico. Ao apresentar os programas SuperLogo 3.0 e Alice 99 a uma criança de oito anos, ela disse ter gostado mais do segundo, possivelmente por causa deste apelo amigável, onde logo de início encontra um mundo mais familiar e atrativo, com objetos 3D lúdicos como personagens de desenho animado. Outro ponto, que atraiu bastante a criança no Alice 99, foi a navegação 3D pelo mundo virtual. Mas depois que a criança satisfez sua atração inicial, ela precisou pensar no que fazer com as personagens, ponto no qual se inicia o valor pedagógico deste ambiente gráfico. A figura 14 apresenta uma tela construída com Alice 99.



Fig. 14 – Exemplo de tela construída no ambiente Alice 99

O programa escolheu uma linguagem de *script* interpretada, orientada a objetos, interativa e de código aberto chamada Python para permitir a criação de programas no ambiente Alice. Esta linguagem foi modificada neste projeto para atender melhor aos objetivos do mesmo.

A utilização de eventos que disparam as animações codificadas é outro ponto positivo deste projeto. O autor pode codificar para os objetos, animações em

resposta a eventos disparados pelo *mouse* e teclado, além do evento de início do mundo *World Start*.

As animações no Alice são baseadas em tempo real, permitindo que o autor defina o tempo em segundos, durante os quais a animação deva se realizar. Desta maneira as animações não são executadas rápido de mais em máquinas velozes, nem de forma muito lenta em máquinas de menor performance.

Alice também não suporta a técnica de animação conhecida por *keyframe*, que corresponde à execução de *frames* de um mesmo modelo 3D, *frames* estes criados pelo animador com o mesmo número de polígonos e vértices. Segundo Conway (1997), os *scripts* do programa Alice são interativos, e os sistemas que utilizam esta técnica de animação geralmente não são.

Alice não fornece recursos para modelagem 3D, pelo contrário, incentiva o uso de modelos criados por ferramentas externas dedicadas a esta finalidade, como por exemplo a ferramenta Teddy que é integrada com a versão do Alice 99. Este modelador, segundo Igarashi, Matsuoka e Tanaka (1999), permite modelar facilmente a partir de esboços livres desenhados em 2D, procurando não cair na dificuldade e tédio encontrados geralmente nas ferramentas do tipo Desenho Apoiado por Computador ou CAD (*Computer Aided Design*) de modelagem.

CAPÍTULO 3

PROPOSTA DE UM SISTEMA DE AUTORIA PARA CRIAÇÃO DE AMBIENTES IMERSIVOS: O SISTEMA FADAS

*“Pretinha e Branquinha são duas lindas fadinhas.
Pretinha é tal qual seu nome diz:
uma garota cor de jabuticaba.
Quando ri, aparecem seus dentes de pérola.
Branquinha é loira de olhos pretos,
face corada. Entre as duas,
não há como escolher,
são ambas dois amorecos”*

*Lúcia Pimentel Góes**

* GÓES, L.P. Pretinha e Branquinha. Aparecida: Editora Santuário, 1991.

Apesar do grande potencial que as tecnologias de Realidade Virtual têm, em particular as imersivas, a sua aplicação na área de educação ainda não se concretizou. A complexidade tecnológica apresenta dificuldades no seu emprego, atrasando a entrada desde rico ambiente interativo nas escolas.

Como criar uma aplicação em um ambiente imersivo de múltiplas projeções como o da CAVERNA Digital (ZUFFO et al., 2001) ? O que é necessário ? Primeiro é preciso criar objetos tridimensionais e sintetizar cenas com eles, criar cenários e personagens virtuais que se movam como no mundo real ou na sua proximidade. Depois a projeção da cena virtual deve ser também projetada nas demais paredes de projeção da CAVERNA Digital, cada projeção com um ponto de vista diferente do observador ou usuário. As projeções também precisam ser estereoscópicas para o usuário sentir se realmente dentro de um mundo virtual, percebendo a profundidade dos objetos que “saltam” do plano de projeção aproximando-se dele ou se afastando para além do plano de projeção e do alcance do usuário. Um mundo virtual também precisa de som, ou melhor, de som tridimensional, que tem posição e orientação no espaço. E como o usuário, dentro da CAVERNA irá interagir com os objetos ? Como ele irá caminhar ou navegar no mundo virtual ? Como ele irá manipular e modificar os objetos ? E depois que cada tecnologia for dominada, com qual linguagem de programação o usuário-autor irá definir as regras, comportamentos e respostas aos eventos disparados pelos objetos interagindo entre si e com o usuário ? Todas estas questões envolvem tecnologias complexas e diversas, cada uma com suas particularidades, APIs, alternativas de *software* e *hardware*. Isto tudo parece muito longe do pesquisador comum, não especialista, quanto mais de um professor ou estudante do ensino fundamental. Deve ser coisa de “rato de laboratório” de Realidade Virtual.

A solução proposta por esta pesquisa consiste no encapsulamento de cada tecnologia e recurso de Realidade Virtual, disponibilizando os poderosos recursos deste ambiente para o usuário comum, não especialista, através de uma linguagem de programação acessível, com comandos simples, com a qual o usuário criará e manipulará mundos virtuais.

Propõe-se então o sistema FADAS como um ambiente de autoria que permita a criação e manipulação de mundos virtuais em ambientes imersivos por usuários não especialistas.

Para tal, o sistema FADAS propõe aliar o benefício para a aprendizagem que um *software* do tipo linguagem de programação proporciona, com a possibilidade de criação e manipulação de mundos virtuais em ambientes imersivos de múltiplas projeções.

O público alvo do sistema FADAS são crianças já alfabetizadas da 1ª a 8ª série do ensino fundamental. O sistema também pode ser útil para usuários de outras faixas etárias no ensino de linguagens de programação e experiências em ambiente imersivo, graças a característica genérica de sua linguagem de programação e por remover as complexidades das tecnologias envolvidas.

O sistema FADAS tem duas categorias diferentes de usuários: o “usuário-autor” que projeta e implementa mundos virtuais através da linguagem Fadas, definindo o comportamento dos objetos e reações a eventos; e o “usuário-explorador” que navega e interage com o mundo virtual construído pelo “usuário-autor”.

O ambiente imersivo de múltiplas projeções utilizado no projeto do sistema FADAS é a CAVERNA Digital (ZUFFO et al., 2001) da Escola Politécnica da Universidade de São Paulo. Com o sistema FADAS na CAVERNA Digital, o “usuário-autor” constrói o cenário de seu mundo virtual, adicionando modelos tridimensionais de construções, veículos, utensílios domésticos, árvores, montanhas, animais, monstros, ou qualquer outro objeto que o “usuário-autor” imagine e projete, construído e disponibilizado por um especialista em modelagem. O “usuário-autor”, além de poder utilizar modelos tridimensionais complexos e animados construídos por especialistas, pode ele próprio criar sólidos simples, texto tridimensional ou importar objetos da biblioteca de objetos e sons, biblioteca fornecida pelo sistema FADAS.

Depois do cenário pronto e das personagens definidos, o “usuário-autor” pode criar um programa para controlar o mundo virtual, manipulando objetos e codificando respostas para as interações entre os objetos virtuais e também interações envolvendo os usuários reais, localizados dentro da CAVERNA Digital, mas que

também estão presentes no mundo virtual. O “usuário-autor” pode tanto construir o seu mundo virtual utilizando a aplicação Console do sistema FADAS instalada em um computador pessoal, localizado fora do ambiente imersivo, ou pode optar por construir o mundo virtual trabalhando dentro do ambiente imersivo da CAVERNA Digital, através da aplicação Console instalada em um computador de dimensões menores como um Assistente Digital Pessoal (PDA - *Personal Digital Assistant*) ou palm-size. A aplicação Console do sistema FADAS é a principal interface do usuário-autor e está descrita no capítulo 4.

O sistema FADAS também pode ser utilizado em ambientes não imersivos, como o de computadores pessoais, fornecendo um ambiente mais propício ao “usuário-autor” que precisa de maior tempo para executar suas atividades, particularmente a de escrever e testar seus programas na linguagem Fadas. Este ambiente de computador pessoal tem um custo muito mais acessível e já é encontrado nas escolas. Incluindo-se um *joystick* no computador, o ambiente se torna mais agradável para o “usuário-explorador” navegar e interagir com um mundo virtual. Se houver interesse em aumentar um pouco a imersão neste ambiente, uma opção é a utilização de monitores de vídeo com maior tamanho que o comum, por exemplo com 17 ou 19 polegadas, e acrescentando óculos obturadores para o “usuário-explorador” sentir o efeito de profundidade experimentado na visualização de imagens estereoscópicas geradas pelo sistema FADAS.

Recomenda-se atividades de projeto e implementação de jogos no sistema FADAS, pois o sistema apresenta vários conceitos de jogos de computador, remove as complexidades das tecnologias envolvidas, permite o compartilhamento de um mesmo mundo virtual e está sendo projetado especialmente para as crianças.

A figura 15 apresenta a tela de abertura do sistema FADAS, cujo protótipo foi desenvolvido nesta pesquisa. Nesta tela são mostradas ao usuário as fadas Pretinha e Branquinha, personagens do livro infantil Pretinha e Branquinha de Lúcia Pimentel Góes (1991), que inspiraram o nome do sistema. Os modelos tridimensionais foram desenhados por colegas do laboratório LSI, especialistas em modelagem 3D.



Fig. 15 – Tela de abertura do protótipo do sistema FADAS

3.1 Linguagem Fadas

O sistema FADAS propõe uma linguagem de programação também denominada Fadas, para que o “usuário-autor” possa criar e manipular objetos em mundos virtuais. A linguagem Fadas é baseada em objetos que podem executar ações ou terem suas características alteradas. Ela foi projetada com o objetivo de ser o mais simples possível para ser compreendida e dominada por crianças.

Utilizando-se esta abordagem orientada a objetos, a linguagem Fadas busca fornecer comandos simples, com conceitos próximos do que é encontrado no mundo real, facilitando abstrações.

O encapsulamento das complexidades das tecnologias envolvidas no ambiente de Realidade Virtual é sempre um objetivo da linguagem de programação Fadas e seus comandos propostos. Por exemplo, os comandos *vira*, *inclina* e *angula*, descritos com maior detalhe no capítulo 4, executam rotações nos eixos do sistema de coordenadas locais do objeto, y, x e z. Mas o usuário comum, não especialista, não precisa conhecer estes eixos ou estes tipos de rotação. No mundo real não existe a necessidade de se lidar com coordenadas. Através destes comandos simples, o sistema FADAS permite que o usuário faça rotações nos objetos tridimensionais de maneira intuitiva.

Procurou-se disponibilizar recursos básicos de uma pequena linguagem de programação, preocupando-se em não torná-la de difícil aprendizagem e uso por parte de crianças e iniciantes em programação. Por exemplo, ao invés de se utilizar o termo variável para armazenar um valor em memória, a linguagem propõe uma abstração mais próxima do mundo real, a criação de *caixas* para o “usuário-autor” guardar os valores que necessite para construir a lógica de seu programa. Também visando facilitar o seu uso por crianças, a linguagem Fadas não faz distinção entre comandos escritos com palavras maiúsculas ou não, acentuadas ou não. A figura 16 apresenta uma tela do sistema FADAS incluindo a janela “Comando”, onde os comandos são digitados e a janela “Objetos” que lista os objetos do mundo virtual.

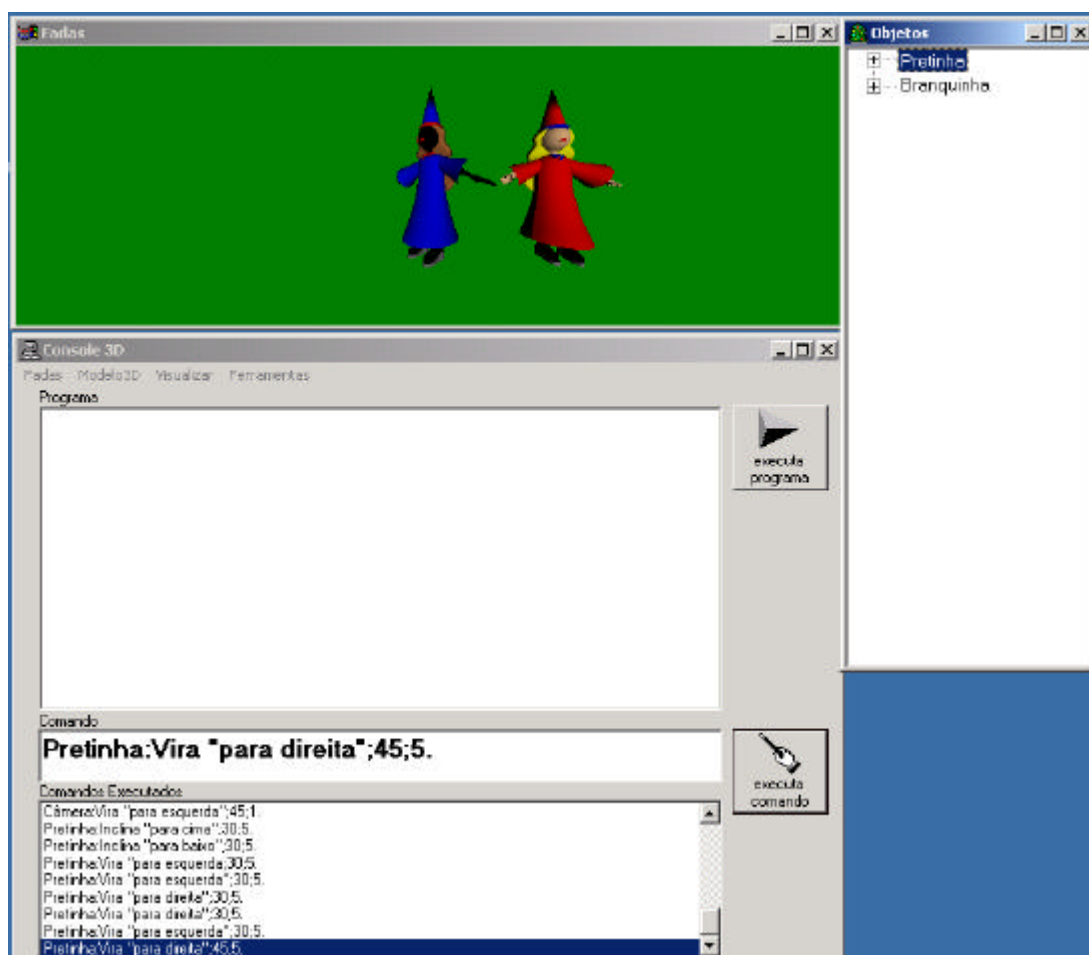


Fig. 16 – Telas do sistema FADAS: Mundo Virtual e Console

3.2 Manipulação de Objetos Tridimensionais

O sistema FADAS permite que o usuário movimente objetos com seis graus de liberdade, efetuando deslocamento e rotação relativa através dos comandos *move*, *vira*, *inclina* e *angula*. Para a execução destes comandos e de outros, o “usuário-autor” pode definir uma duração sem segundos, criando assim animações.

Outros comandos permitem a alteração das dimensões dos objetos, a cor refletida e a opacidade do objeto, ou mesmo tornar o objeto invisível ou visível.

Algumas características físicas foram implementadas e disponibilizadas para o usuário, como velocidade, aceleração e coeficiente de atrito.

3.2.1 Navegação em Ambiente Imersivo

O “usuário-explorador”, para navegar no ambiente imersivo de projeções múltiplas, pode utilizar um teclado ou um *joystick*, preferencialmente sem fio, sendo que este último é o mais natural e tem a vantagem de familiaridade com os jogos de video game. O uso de outros dispositivos acoplados ao *joystick*, como por exemplo um *skate* disponível na CAVERNA Digital, pode tornar a navegação mais natural, facilitando e agilizando esta atividade. Um teclado sem fio ou mesmo um computador do tipo *palm-top* também poderiam ser utilizados neste ambiente para navegação, porém diminuiria um pouco o alto nível de imersão encontrado na CAVERNA Digital.

A navegação é implementada em primeira pessoa, associando-se os olhos do “usuário-explorador” com uma câmera, e aplicando as mesmas ações de movimento e rotação aplicáveis aos objetos tridimensionais.

O sistema FADAS permite que o “usuário-autor” crie diversas câmeras no mundo virtual, visando disponibilizar diferentes pontos de vista ao “usuário-explorador”. As câmeras recebem nomes podem então ser manipuladas, ativadas e eventualmente associadas a outros objetos tridimensionais, neste caso o sistema manterá a câmera sempre alinhada com o objeto associado.

Outro recurso interessante disponibilizado pelo sistema FADAS para a navegação, é a possibilidade de se atribuir uma velocidade à câmera, com a flexibilidade de se alterar ou não a direção da câmera quando o “usuário-explorador” efetuar uma rotação na câmera além de ajustar a intensidade da velocidade de acordo

com o interesse do “usuário-explorador”. Desta maneira, o usuário explorador pode “passar” no mundo virtual, como que “guiando um veículo espacial” ou “voando como uma fada”.

No ambiente de multi-projeção da CAVERNA Digital, o usuário além de navegar com um *joystick* ou teclado, pode simplesmente caminhar fisicamente pelo ambiente e interagir com objetos próximos, neste caso o sistema de rastreamento detecta a nova posição do usuário e movimenta a câmera de acordo com o novo ponto de vista.

3.2.2 Manipulação de Objetos em Ambiente Imersivo

Para manipular objetos do mundo virtual no ambiente imersivo da CAVERNA Digital, o “usuário-explorador” pode utilizar os botões do *joystick* e uma “varinha mágica”, aproveitando-se a metáfora das varinhas utilizadas por fadas e mágicos. Esta “varinha mágica” é um dispositivo de controle virtual, associado a um dispositivo de rastreamento com posição e orientação no espaço. Através dela o usuário pode “tocar” e selecionar objetos, e disparar ações ou alterar características dos objetos com os botões do *joystick*, dependendo da criatividade do programa em execução criado pelo “usuário-autor”.

Um recurso que pode ser objeto de futuras pesquisas, visando ampliar o poder de interação do usuário com o mundo virtual em ambiente imersivo, é a interpretação de comandos de voz pelo sistema FADAS. Esta seria uma forma bastante natural e eficaz de se manipular o mundo virtual. Outros dispositivos de interação das tecnologias de Realidade Virtual, como luvas de dados e dispositivos *haptics*, também podem ser pesquisados visando fornecer outras opções de interface para as aplicações desenvolvidas no sistema FADAS.

3.3 Criação de Objetos Tridimensionais

O sistema FADAS permite a criação, adição e construção de diversos tipos de objetos tridimensionais. Cada objeto possui uma posição central e orientação no mundo virtual, permitindo a sua manipulação através dos comandos da linguagem Fadas.

Os objetos adicionados ao mundo virtual podem ser posicionados e orientados de duas maneiras: na posição e orientação de um objeto pré-existente, como, por exemplo, uma das fadas, ou na posição central do mundo virtual alinhado com o eixo z das coordenadas fixas globais. A descrição do sistema de coordenadas global do mundo virtual e do sistema de coordenadas local dos objetos, é descrita no item “4.10 - Movimento e Rotação Relativa” do capítulo “4 - Implementação”.

3.3.1 Hierarquia entre Objetos

No sistema FADAS, os objetos tridimensionais são organizados de maneira hierárquica, visando facilitar a manipulação dos mesmos. Quando um comando é executado sobre um objeto, o mesmo é replicado para cada “objeto filho” associado.

Cada objeto pode ter vários “objetos filhos”. Cada “objeto filho” tem um “objeto pai” e pode ter, por sua vez, vários “objetos filhos”, e assim por diante. Cada objeto ou sub-objeto tem um nome atribuído, que pode ser alterado pelo usuário. A alteração na hierarquia dos objetos também pode ser feita, agrupando-se sub-objetos em um “objeto pai”, permitindo assim a manipulação de todo o grupo de objetos abaixo do “objeto pai”.

3.3.2 Biblioteca de Objetos e Sons

O sistema FADAS fornece ao usuário um conjunto de objetos tridimensionais como personagens, animais, construções, plantas, veículos, móveis, utensílios domésticos e outros.

O sistema FADAS não é um modelador de objetos tridimensionais, e necessita, portanto, que modelos 3D sejam criados por especialistas e importados pelo sistema FADAS. Os formatos ASE e MD2 são suportados para importação, incluindo modelos com texturas 2D de imagens nos formatos BMP, PCX e PPM e modelos com animações criadas com a técnica de *keyframes*. O número de modelos 3D disponibilizados na Internet, com ou sem custo, com ou sem animação, é muito grande e não para de crescer, podendo ser utilizados para montar os cenários e personagens dos mundos virtuais. Modelos 3D criados com fins didáticos também são uma boa idéia para serem explorados em ambiente imersivo de múltiplas projeções através do sistema FADAS.

Alguns arquivos de som digitalizado e música sintetizada também são fornecidos para a criação de mundos virtuais, como por exemplo sons de explosão, tiro, disparo de laser, campainha, raquete de tênis batendo em uma bolinha e músicas. O “usuário-autor” pode também ele mesmo gerar sons e músicas e importá-los para um mundo virtual do sistema FADAS. Os formatos WAVE e MIDI são suportados para importação.

3.3.3 Criação de Pontos, Linhas, Polígonos, Sólidos e Textos Tridimensionais

O sistema FADAS permite ao usuário a criação de pontos, linhas, ou melhor, segmentos de linha formados por pontos interligados e polígonos. Os polígonos devem ser convexos e podem ser construídos de três maneiras: um conjunto de triângulos, um conjunto de quadriláteros ou um conjunto de vértices.

O sistema também permite ao usuário a criação e manipulação de textos tridimensionais e sólidos básicos do tipo esfera, paralelepípedo, cubo, cilindro, cone e anel, a partir da definição dos parâmetros da geometria do objeto.

3.3.4 Criação de Fontes de Luz e Som Tridimensional

O sistema FADAS permite que o “usuário-autor” adicione e manipule outras fontes de luz direcionais ou posicionais no mundo virtual, visando alterar a luminosidade das cenas.

O sistema FADAS permite que o “usuário-autor” crie e manipule fontes de som tridimensionais, com posição e eventualmente velocidade e orientação no espaço virtual. O “usuário-autor” pode assim controlar a intensidade de som à medida que o “usuário-explorador” se aproxime ou se distancie de uma fonte de som tridimensional.

3.4 Colaboração

O sistema FADAS permite que vários “usuários-exploradores” compartilhem um mesmo mundo virtual, ou mesmo que vários “usuários-construtores” construam mundos virtuais em conjunto, através da adição e manipulação de objetos tridimensionais.

Cada objeto guarda informações sobre o seu criador e só pode ser removido por ele ou pelo administrador do sistema. Somente um usuário por vez pode manipular um objeto.

Cada usuário configurado e habilitado pelo administrador, tem um objeto tridimensional associado, ou avatar, e pode navegar e interagir com o mundo virtual. Quando um “usuário-explorador” navega, ele visualiza os demais usuários, em tempo real, através dos objetos associados, permitindo assim a colaboração entre os usuários, ou mesmo a presença de um usuário remoto interagindo com usuários localizados dentro da CAVERNA Digital. A comunicação com o usuário remoto, neste caso, pode ser feita através de um microfone sem fio escondido na CAVERNA Digital e controlado por algum *software* de conversação em tempo real.

Depois que o mundo virtual estiver construído, um programa criado pelo “usuário-autor” pode ser executado, permitindo assim, por exemplo, que um jogo com a participação de vários usuários remotos seja jogado ou que outra atividade colaborativa qualquer, controlada pelo programa escrito pelo “usuário-autor”, seja realizada.

3.5 Comparação entre os ambientes virtuais de autoria

Esta comparação entre ambientes virtuais de autoria, apresentada na tabela I, foi feita com base nas versões dos *softwares* instalados e respectivos sistemas de ajuda (*help*). Ela permite uma melhor visão ao leitor sobre as características de cada ambiente. A listagem, com a descrição dos comandos da linguagem Fadas, pode ser consultada no anexo A desta dissertação.

Tabela I – Comparação entre ambientes virtuais de autoria

Descrição	FADAS	Alice 99	SuperLogo 3.0	Imagine 1.50
Aplicação de domínio específico ou geral	Geral	Geral	Geral	Geral
Versão em português	Sim	Não	Sim	Sim
Linguagem de programação	Sim	Sim	Sim	Sim
Linguagem baseada em objetos	Sim	Sim	Não	Sim
Programação de eventos	Sim	Sim	Sim	Sim
Desenho tridimensional	Sim	Sim	Sim	Não
Síntese de imagens tridimensionais	Sim	Sim	Não	Não
Importação de modelos 3D	Sim	Sim	Não	Não
Criação de desenhos em 2D	Não	Não	Sim	Sim
Criação de desenhos em 3D	Sim	Não	Sim	Sim
Manipulação de fontes de luz	Sim	Sim	Não	Não
Manipulação de objetos (alteração de propriedades físicas)	Sim	Sim	Não	Sim
Manipulação de câmeras	Sim	Sim	Não	Não
Colaboração entre usuários	Sim	Não	Não	Não
Troca de informações por rede	Não	Não	Sim	Sim
Suporte a ambiente de múltiplas projeções	Sim	Não	Não	Não
Geração de imagens estereoscópicas	Sim	Não	Não	Não
Biblioteca de objetos, cenários e sons.	Sim	Sim	Não	Sim
Suporte a vídeo (MCI)	Não	Não	Sim	Sim
Suporte para músicas e sons	Sim	Sim	Sim	Sim
Som tridimensional	Sim	Não	Não	Não
Animação baseada em <i>script</i>	Sim	Sim	Não	Não
Animação <i>keyframing</i>	Sim	Não	Não	Não
Criação de figuras animadas	Não	Não	Não	Sim
Suporte a comandos de voz	Não	Não	Não	Sim
Deteção de colisão entre objetos	Sim	Sim	Não	Sim

CAPÍTULO 4

IMPLEMENTAÇÃO

*“É precisamente no sentido da apropriação dos meios high tech de produção de imagens que se pode e deve, hoje, considerar o desafio que a poesia propõe à criatividade e à comunicação. A criação de uma infopoesia, uma videopoesia e uma holopoesia é uma parte desse desafio, no prosseguimento da pesquisa dos suportes e dos meios que a poesia visual tem sido até agora. Pesquisa que utilizará, respectivamente, o computador, os sistemas de produção de imagens em vídeo e a holografia e outros meios de alta tecnologia, não para imbecilizar, controlar, explorar, manipular e matar os homens, isto é, nós, mas para aumentar a nossa própria capacidade de criação e de fruição. Desse modo, o fim visual do século XX é a cores – cores geradas eletronicamente, qualitativa e esteticamente diferentes das cores do mundo natural, mas talvez ainda mais distantes das cores químicas da pintura. Estas, as cores da pintura, são estáticas e pesadas. Mas as cores eletrônicas são instáveis e leves, porque são luz, provenientes de um emissor de raios catódicos que se transforma em fóton, como no vídeo, ou são luz homogeneizada, na holopoesia. Trabalhar com meios desmaterializados, que são própria energia luminosa, criando padrões de percepção colorida que de outro modo não poderão ser produzidos e que têm a sua sintaxe própria: sintaxe que é tarefa do poeta inventar e desenvolver, isto estou certo, é o desafio mais fascinante da atual poesia visual.”**

Ernesto M. de Melo e Castro

* CASTRO, E.M.M. *O Fim Visual do Século XX e Outros Textos Críticos*. São Paulo, Editora da Universidade de São Paulo, 1993.

Neste capítulo são descritos detalhes de projeto e implementação do sistema FADAS, com destaque aos aspectos técnicos internos do projeto, que permitiram criar a linguagem de programação Fadas encapsulando as complexidades dos recursos e tecnologias de um ambiente imersivo de múltiplas projeções.

4.1 Arquitetura

O sistema FADAS, cuja arquitetura é apresentada na figura 17, é composto de duas aplicações: a aplicação Mundo Virtual que controla o mundo virtual, desenvolvida na linguagem de programação Microsoft[®] Visual C++ e a aplicação Console, que fornece uma interface para o usuário enviar comandos e escrever programas, desenvolvida na linguagem de programação Microsoft[®] Visual Basic.

A aplicação Mundo Virtual é composta de uma série de classes internas responsáveis pelas diversas funcionalidades do sistema. Isto facilita a portabilidade do sistema, assim como o uso de bibliotecas ANSI (*American National Standards Institute*). Classes de manipulação de cadeia de caracteres, listas, vetores e matrizes, entre outras, foram desenvolvidas visando facilitar a portabilidade do sistema para outras plataformas de *hardware* e *software*.

A biblioteca gráfica OpenGL é utilizada para sintetizar as cenas tridimensionais. Esta é uma biblioteca largamente utilizada, disponível em várias plataformas de sistema operacional e suportada pelas placas controladoras e aceleradoras gráficas. Assim a portabilidade é facilitada e o *hardware* gráfico é bem aproveitado.

A biblioteca Windows Sockets fornece os recursos de comunicação entre a Aplicação Mundo Virtual e a Aplicação Console, permitindo assim que elas sejam instaladas em computadores diferentes. Na configuração do sistema para a CAVERNA digital, o sistema também utiliza Windows Sockets para a comunicação entre as diversas instâncias da Aplicação Mundo Virtual, responsáveis pelas projeções. Sockets também é um padrão disponível em várias plataformas de sistema operacional.

As bibliotecas Microsoft® Direct Input e Microsoft® Direct Audio também foram encapsuladas em classes dedicadas e foram adotadas por fornecer acesso aos recursos de *hardware* com a devida performance necessária para aplicações de jogos interativos, como é o caso deste sistema.

A Aplicação Console fornece um painel de controle para o usuário visualizar informações sobre o mundo virtual, alterar configurações, e enviar comandos e programas a serem executados pela Aplicação Mundo Virtual. Ele foi desenvolvido na linguagem MS Visual Basic por causa de sua alta produtividade.

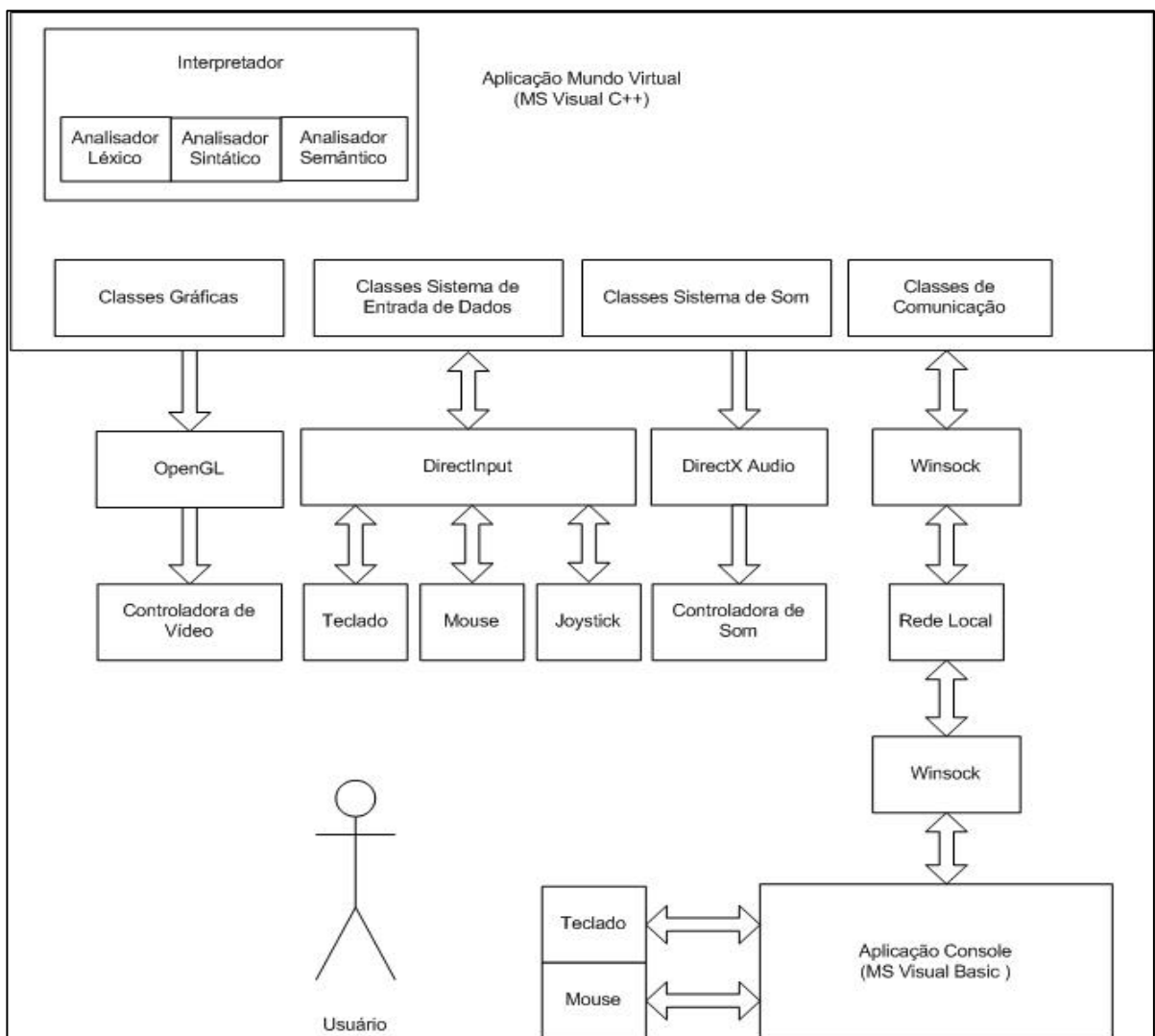


Fig. 17 – Arquitetura do sistema FADAS

4.2 Classes Internas

O diagrama de classes na notação UML (*Unified Modeling Language*) das classes desenvolvidas internamente na aplicação Mundo Virtual é apresentado nas figuras 18a e 18b. A seguir descrevemos o uso de algumas classes da aplicação Mundo Virtual, citando o nome delas entre parênteses. Utilizou-se o seguinte padrão de nomenclatura, para facilitar a manutenção e legibilidade do código fonte: todo nome de classe é iniciado com a letra C maiúscula.

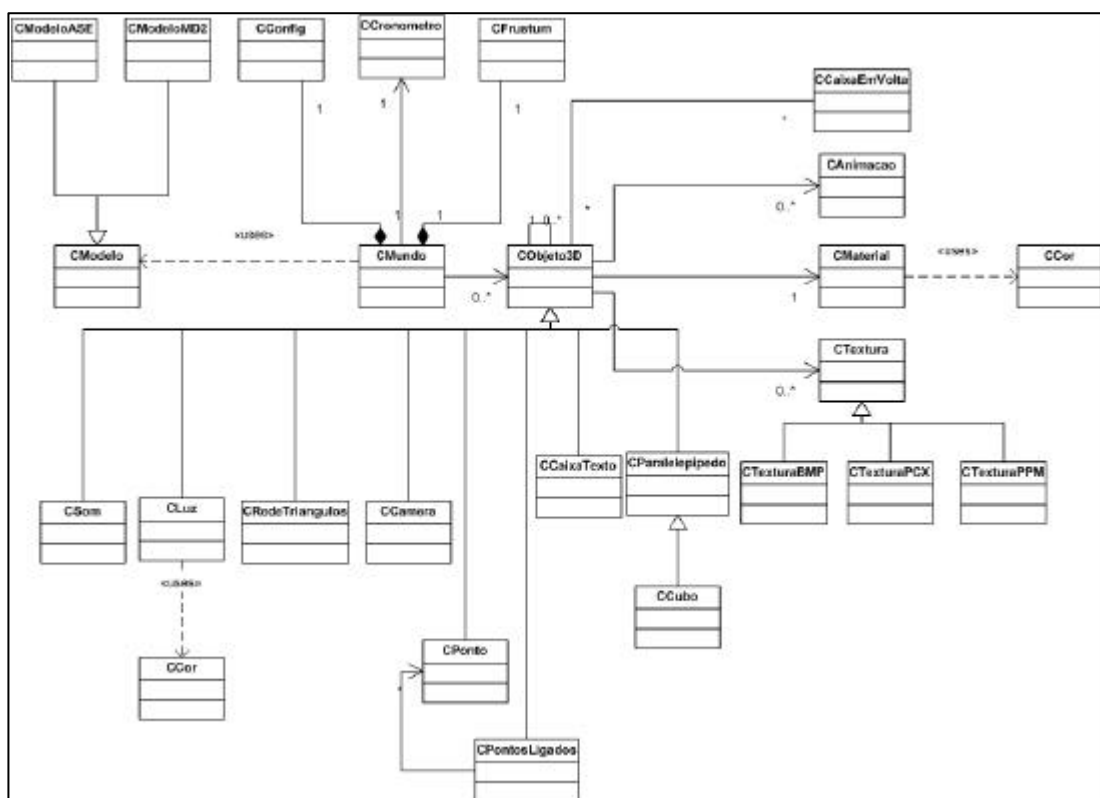


Fig. 18a – Classes internas do sistema FADAS

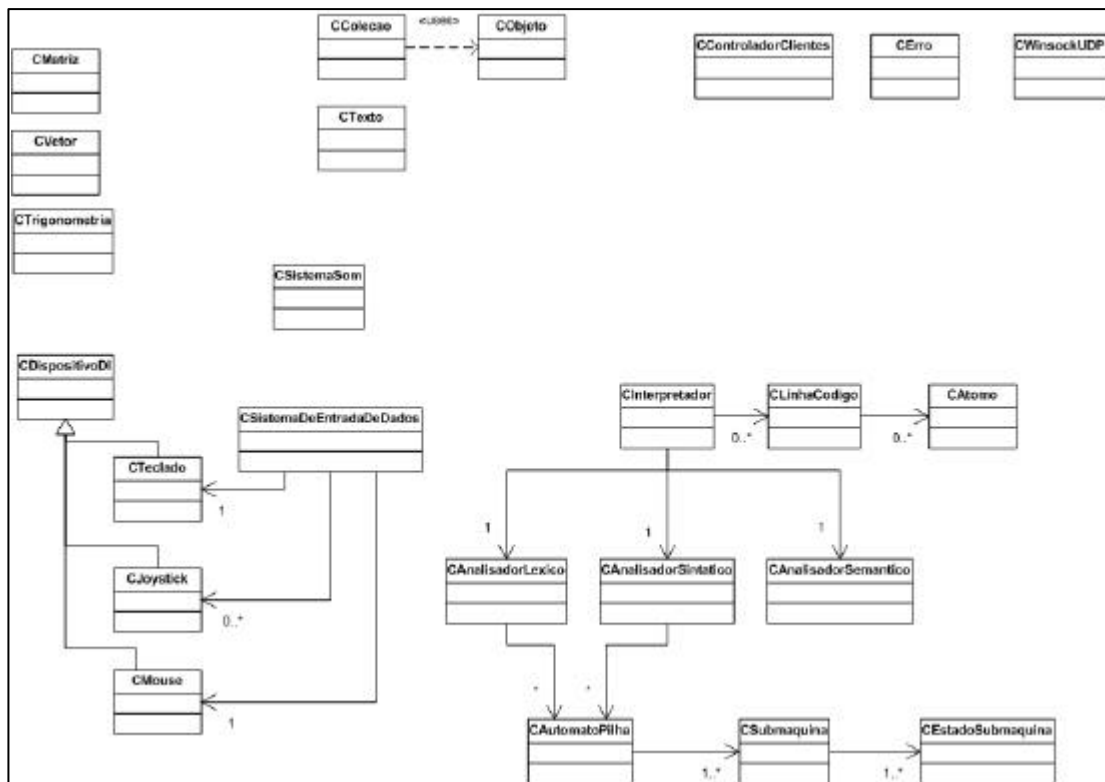


Fig. 18b – Classes internas do sistema FADAS

O mundo virtual (CMundo) possui uma lista de objetos (CObjeto3D). A cada objeto adicionado ao mundo, uma nova instância da classe CObjeto3D é criada e acrescentada na lista. O Mundo também tem uma lista de fontes de luz (CLuz), uma lista de câmeras (CCamera) e uma lista de sons (CSom). Estas listas são implementadas pela classe CCoolecao. A classe CCamera é a abstração dos olhos do usuário, permitindo a navegação no mundo virtual, a classe CLuz simula as luzes do mundo real e a classe CSom representa os sons sintetizados ou digitalizados que o usuário carrega no mundo virtual, podendo ser ou não tridimensional.

As luzes, câmeras e objetos têm um nome que é implementado pela classe CTexto. Esta classe encapsula toda a manipulação de cadeias de caracteres, efetuando funções como tamanho da cadeia, concatenação, comparação. A linguagem Fadas não diferencia letras maiúsculas de minúsculas, nem letras acentuadas das não acentuadas. Esta classe CTexto é quem viabiliza esta funcionalidade.

Os objetos podem ser inseridos no Mundo através da classe CModelo responsável por importar modelos 3D. As classes CModeloMD2 e CModeloASE implementam a importação dos formatos MD2 e ASE. Os objetos 3D são implementados pela classe CRedeTriangulos, que mantém a lista de vértices, lista de triângulos, coordenadas de textura e normais, com o auxílio da classe CVetor.

Todo objeto (CObjeto3D) possui um material (CMaterial) que define a cor (CCor), ou melhor, as luzes vermelha, verde e azul refletidas pelo objeto. Um objeto pode possuir uma ou mais texturas (CTextura). Os formatos de arquivo de imagens 2D implementados para as texturas foram BMP, PCX e PPM. Um objeto pode possuir várias animações (CAnimação) fornecidas pelo modelo tridimensional.

A classe CWinsockUDP foi criada para abstrair a comunicação entre computadores através da Interface de Programa de Aplicação (API) Winsock e acessar os serviços do protocolo de transporte do TCP/IP (*Transmission Control Protocol/Internet Protocol*).

A classe CSistemaDeEntradaDeDados é responsável pela interatividade da ferramenta com o usuário, e abstrai dispositivos como Teclado, *Mouse* e *Joystick*.

A classe CSistemaDeSom é responsável por tocar música e sons pré-gravados.

4.3 Hierarquia entre objetos

No sistema FADAS, cada objeto tridimensional do mundo virtual pode ter vários “objetos filhos”. Cada “objeto filho” tem um “objeto pai” e pode ter vários “objetos filhos”, e assim por diante.

Quando o usuário solicita que um comando para um objeto tridimensional, como, por exemplo, a atribuição de uma cor, o sistema FADAS altera a cor do objeto e repete a operação recursivamente para cada objeto de sua coleção de “objetos filhos”. No exemplo de código 1 em linguagem C++, o sistema altera recursivamente a cor da luz difusa refletida por um objeto tridimensional.

Exemplo de código 1 – Hierarquia entre objetos (linguagem C++)

```

void CObeto3D::DefineMatLuzDifusa(CCor &oCor)
{
    LONGO_SEM_SINAL ulFilhos;
    CObeto3D *pFilho;
    // Aplica para filhos se houver
    for (ulFilhos = 0; ulFilhos < mFilhos.ConsultaTotalObjetos(); ulFilhos++)
    {
        pFilho = (CObeto3D *) mFilhos.PegaObjeto(ulFilhos);
        pFilho->DefineMatLuzDifusa(oCor);
    }
    mMaterial.DefineMatLuzDifusa(oCor);
}

```

4.4 Plataforma de Desenvolvimento

4.4.1 Hardware

Recomenda-se a seguinte configuração mínima para o sistema FADAS na plataforma Intel x86:

- processador Pentium III de frequência de 500 MHz.
- 128 Mbytes de memória RAM (*Random Access Memory*).
- controladora de vídeo com suporte ao modo 800 x 600 e 16 bits de cor.
- joystick do tipo game pad (opcional).
- espaço em disco rígido disponível de 200 Mbytes

O desempenho do sistema pode ser bastante melhorado elevando-se a capacidade de processamento do computador e utilizando-se uma controladora de vídeo que acelere funções gráficas.

4.4.2 Software

O sistema FADAS é suportado na plataforma Microsoft® Windows e requer a biblioteca de execução (*run-time*) Microsoft® DirectX 8 ou superior e Microsoft® Visual Basic 6.0.

O sistema foi desenvolvido com os seguintes componentes de *software*:

- Sistema Operacional Microsoft® Windows 2000 Server (com *service pack 2*).
- Microsoft® Visual C++ .Net.
- Microsoft® Visual Basic 6.0 (com *service pack 5*).

- Microsoft® DirectX 8 SDK.

4.5 Interfaces com o usuário

O console do sistema FADAS corresponde à interface de usuário para controlar o mundo virtual. Nesta interface o usuário recebe as informações de quais objetos fazem parte do mundo virtual, escreve comandos e programas, adiciona objetos e configura o ambiente.

A tela de comandos do console, apresentada na figura 19, é principal interface de programação. Ela é formada por três janelas: Comando, Resposta e Programa. Sendo que ao abrir a tela, somente a janela “Comando” é mostrada. As janelas “Programa” e “Resposta” são apresentadas ou escondidas de acordo com a vontade do usuário, através do menu Visualizar.

O “usuário-autor”, para enviar um comando ao mundo virtual, digita o texto do comando na janela “Comando” e clica no botão de comando “Executa Comando”.

Na janela “Resposta” o sistema escreve informações para o usuário ler, como, por exemplo, os comandos enviados e repostas a perguntas solicitadas através do comando Pergunta.

Na janela “Programa” o “usuário-autor” escreve os programas a serem executados no ambiente Fadas. A execução do programa é iniciada quando o usuário clica no botão de comando “Executa Programa” ou “Executa Linha”.

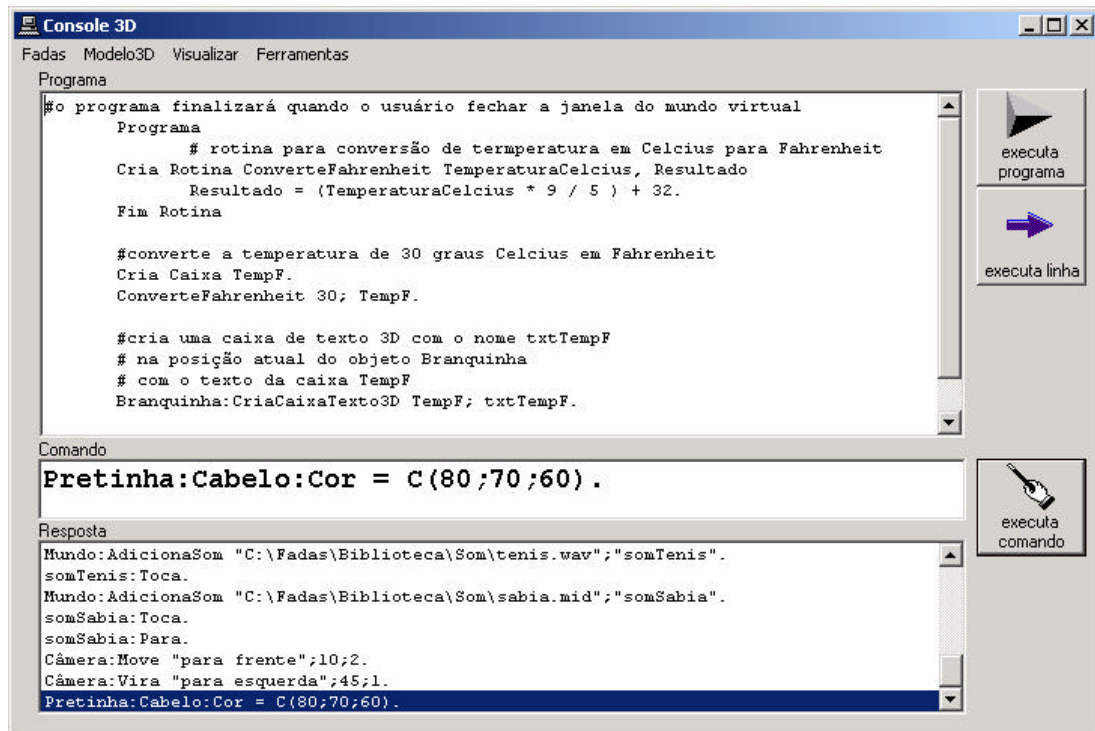


Fig. 19 – Tela de comandos do console

O console possui outras janelas ativadas através do menu “Visualizar” para auxiliar o usuário a criar os comandos Fadas e consultar informações sobre o mundo Virtual. As janelas de auxílio são: Objetos, Animações, Cor, Som, Câmeras e Fontes de Luz.

A janela “Objetos”, apresentada na figura 20, lista os objetos do mundo virtual de maneira hierárquica, permitindo a visualização dos sub-objetos existentes. O usuário pode efetuar um duplo clique no nome do objeto ou sub-objeto para que o sistema escreva o nome dele na janela “Comando”.

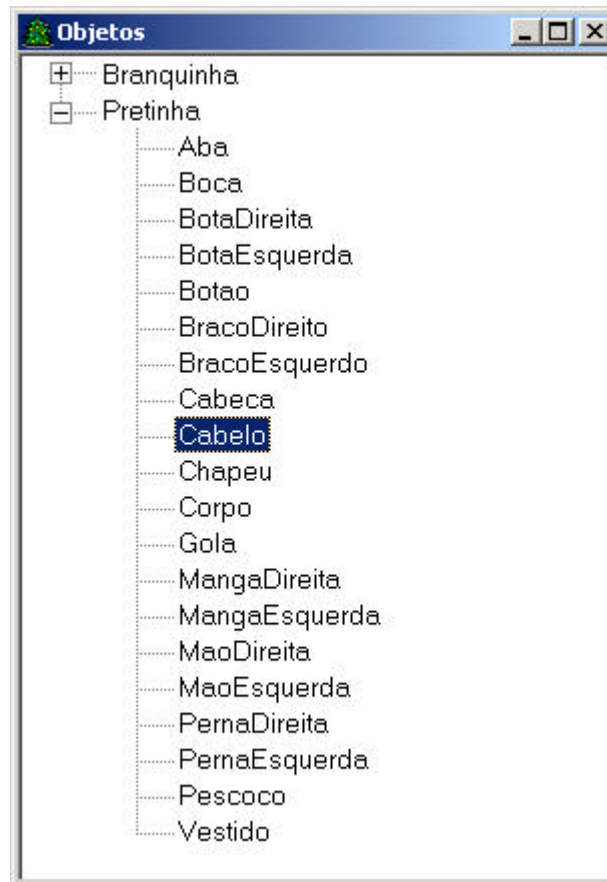


Fig. 20 – Janela Objetos

A janela “Cor”, apresentada na figura 21, permite ao usuário definir uma cor a partir das componentes de luz vermelha, verde e azul. O usuário desliza com o *mouse* variando o valor da componente de luz, entre 0 e 100. Depois de definir os valores das componentes, o usuário pode efetuar um duplo clique no quadrado colorido para que o sistema escreva a cor na janela “Comando”. A sintaxe do valor da cor selecionada é mostrada abaixo do quadrado colorido.

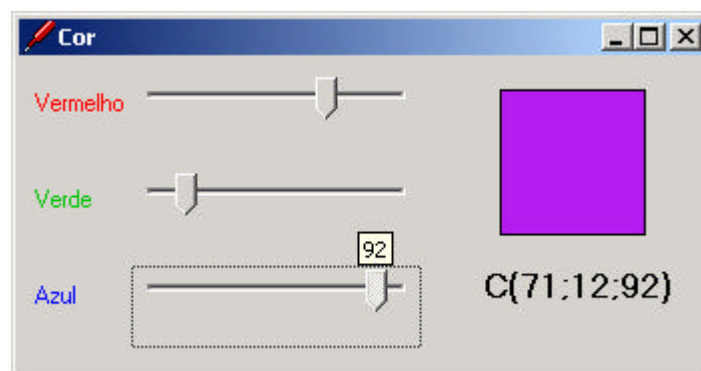


Fig. 21 – Janela Cor

A janela “Animações”, apresentada na figura 22, lista os objetos que possuem animação e o nome de cada uma de suas animações. Quando o usuário efetua um duplo clique no nome da animação, o sistema escreve na janela “Comando” o comando para executar a animação selecionada.

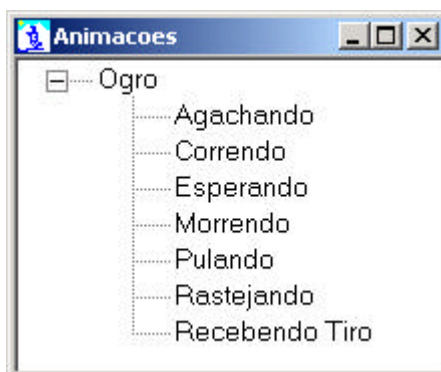


Fig. 22 – Janela “Animações”

A janela “Sons”, apresentada na figura 23, lista os sons carregados no sistema informando o nome de cada um assim como o nome do seu arquivo. Quando o usuário efetua um duplo clique em um som, o sistema escreve na janela “Comando” o comando para reproduzir o som.

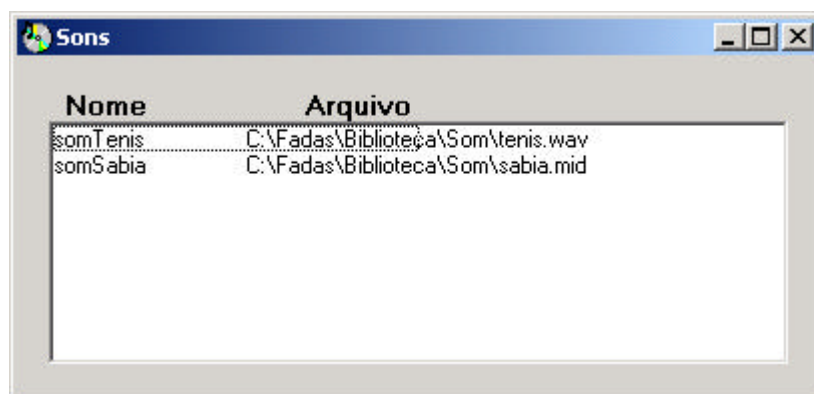


Fig. 23 – Janela “Sons”

A janela “Câmeras”, apresentada na figura 24, lista as câmeras definidas no mundo virtual. Um duplo clique no nome da câmera define-a como a câmera ativa. Através de botões de comando, o usuário pode executar métodos de movimento e rotação na câmera ativa.

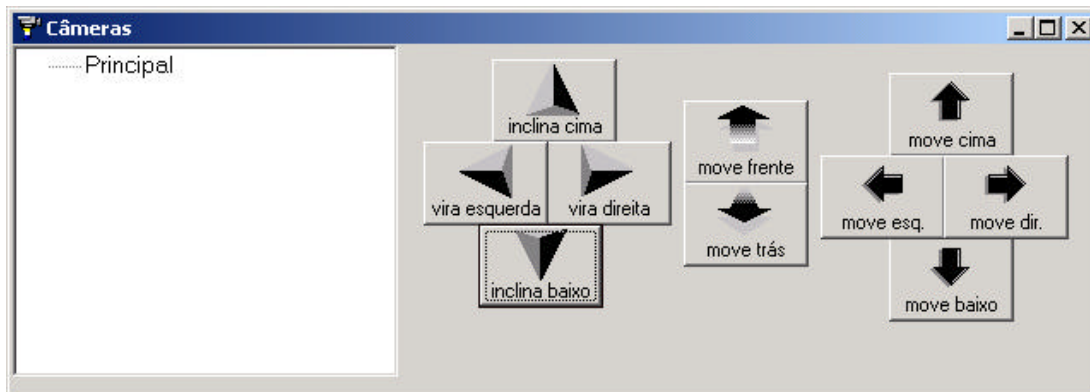


Fig. 24 – Janela “Câmeras”

A janela “Fontes de Luz”, apresentada na figura 25, lista as fontes de luz definidas para o sistema. O usuário pode definir novos valores para as componentes vermelho, verde e azul de luzes ambiente, difusa e especular.



Fig. 25 – Janela “Fontes de Luz”

4.6 Linguagem Fadas

O sistema FADAS, visando oferecer maiores recursos para o controle do mundo virtual, disponibiliza a linguagem de programação Fadas para o “usuário-autor”. Esta linguagem permite a programação dos comportamentos e interações entre os objetos que compõem o mundo virtual, levando vida ao ambiente.

A utilização do sinal de ponto-e-vírgula para separar parâmetros, o ponto final para finalizar um comando, a vírgula como separador decimal de um número foram adotados seguindo a orientação dos especialistas em linguagens de programação Marcotty; Ledgard (1986), ambos aconselham que a linguagem de programação deve tentar manter a utilização de espaços e pontuação de uso costumeiro pelo programador, a fim de evitar erros desnecessários.

4.6.1 Sintaxe, Semântica e Regras Propostas

Os caracteres permitidos na linguagem e o reconhecimento dos átomos da linguagem estão descritos formalmente na notação Wirth (José Neto, 1987), no apêndice I desta dissertação. Um átomo é o menor elemento com significado, reconhecido pelo interpretador da linguagem Fadas.

Inicialmente foi definida uma sintaxe para comandos básicos a serem enviados ao mundo Fadas. Desta maneira se viabilizou os primeiros testes com usuários utilizando o protótipo do sistema. Somente dois tipos de comandos foram definidos: uma ação de um objeto ou a atribuição de um valor a um objeto. A descrição formal, na notação de Wirth, da gramática dos comandos básicos, se encontra no apêndice II desta dissertação.

A seguir a linguagem é descrita informalmente. A descrição formal de sua gramática pode ser encontrada no apêndice III desta dissertação. No próximo item, apresentam-se alguns exemplos de sua utilização. Conforme Marcotty e Ledgard dizem, “Pessoas aprendem por exemplos, e com linguagens de programação isto é particularmente verdadeiro”.

4.6.1.1 Programa

O código de um programa tem início com a palavra reservada “Programa” e termina com as palavras reservadas “Fim Programa” seguida do ponto final, como no exemplo de código 2.

Exemplo de código 2 – Sintaxe Programa (linguagem Fadas)

```
Programa
#comandos e declarações

Fim Programa.
```

Cada linha do programa pode conter uma declaração ou um comando. Existem três tipos de declaração: caixa, rotina e evento, enquanto existem cinco tipos de comandos: “ação, atribuição, condição, laço (*loop*) e chamada de rotina”.

4.6.1.2 Declaração de Caixa

A declaração de uma caixa, apresentada no exemplo de código 3, cria um local de memória para o “usuário-autor” guardar valores. A sua declaração é iniciada com as palavras reservadas “Cria Caixa” seguidas de um identificador. Opcionalmente o autor pode definir um valor inicial para a caixa, através do sinal de igual e de um valor. Ela é finalizada com o ponto final.

Um identificador é um conjunto de caracteres. O primeiro caractere deve ser uma letra e os demais podem ser letras, números ou o símbolo “_” (*underscore*). Espaços não são permitidos em um identificador. As palavras reservadas pelo sistema FADAS não podem ser utilizadas como identificadores. Um identificador não pode ser declarado mais do que uma vez.

Exemplo de código 3 – Declaração de Caixas (linguagem Fadas)

Sintaxe Declaração Caixa: Cria Caixa NomeCaixa [= valor].	
Exemplos:	Cria Caixa ResultadoConta. Cria Caixa NomeEscola = “Canarinho”.

4.6.1.3 Ação de um Objeto

A ação de um objeto, apresentada no exemplo de código 4, consiste de um identificador do objeto, opcionalmente sub-objetos separados pelo sinal de dois pontos, o sinal de dois pontos, um identificador da ação e opcionalmente uma lista de parâmetros separada do identificador por um espaço em branco. O fim de um comando é definido pelo ponto final.

O sistema FADAS, visando facilitar a sua programação, não faz distinção entre letras maiúsculas ou minúsculas nem entre palavras acentuadas ou não. Com o mesmo objetivo, o sistema aceita a contração da preposição para, ao definir, por exemplo, uma direção. Assim é reconhecido o texto referente a direção “pra cima” quanto “para cima”.

Os parâmetros são formados por valores ou identificadores de caixa separados pelo sinal ponto-e-vírgula. Um parâmetro pode ser uma caixa ou um valor. Existem quatro tipos de valores: texto, número, cor e vetor. Um texto é delimitado por aspas. Um número não possui delimitador e pode ter casas decimais após a

vírgula. A cor é formada pela letra C e entre parênteses as componentes de luz vermelha, verde e azul representados por números entre 0 e 100 separados por ponto-e-vírgula. O vetor é formado pela letra V e entre parênteses as coordenadas espaciais x, y e z representadas por números separados por ponto-e-vírgula).

Quando uma caixa é passada como parâmetro para uma ação, ela poder ter o seu conteúdo alterado pela ação. O mesmo ocorre no caso de rotinas de código criadas pelo usuário, cuja sintaxe é descrita no item 4.6.1.7.

Exemplo de código 4 – Ação de um objeto (linguagem Fadas)

Sintaxe Ação:	Objeto[:sub-objetos]:Ação [parâmetros].
Exemplos:	Branquinha:Move “para esquerda”; 10; 1. Pretinha:Cabeça:Inclina “pra cima”; 30; 2.

4.6.1.4 Atribuição

O comando de atribuição, apresentado no exemplo de código 5, associa um valor a uma caixa previamente declarada ou a característica de um objeto. O valor a ser atribuído, localizado à direita do símbolo “=”, pode ser oriundo de uma característica de objeto ou de uma caixa ou de uma expressão numérica.

Uma expressão numérica consiste de operandos separados por operadores aritméticos. Os operandos podem ser identificadores de caixas, identificadores de uma característica de um objeto, um valor numérico ou uma expressão entre parênteses. Os operadores são os símbolos * (multiplicação), / (divisão), + (adição), - (subtração), sendo que os operadores de multiplicação e divisão têm precedência sobre os operadores de adição e subtração.

Exemplo de código 5 – Atribuição (linguagem Fadas)

Sintaxe Atribuição Caixa:	NomeCaixa = valor.
Exemplos:	ResultadoConta = 5. NomeEscola = "São Paulo". VelocidadeAtual = Menino:Velocidade. ResultadoCalculo = (3,5 * ResultadoConta) – 2.
Sintaxe Atribuição Característica:	Objeto[:sub-objetos]:Característica = valor.
Exemplo:	Branquinha:Rosto:Cor = C(80; 30;45). Carro:Velocidade = VelocidadeAtual. Carro:Aceleração = AceleraçãoAtual + 1,5.

4.6.1.5 Comando Condicional

O comando condicional, apresentado no exemplo de código 6, é iniciado com a palavra reservada “Se” seguida de uma comparação numérica e a palavra reservada “Então”. A seguir o “usuário-autor” pode definir um ou mais comandos a serem executados no caso da comparação ser verdadeira até que a palavra reservada “Fim Se” seja encontrada, fechando o comando condicional com o ponto final.

Uma comparação numérica é formada por dois operandos separados por um operador de comparação. Os símbolos dos operadores de comparação são “=” (igual a), “<>” (diferente de), “>” (maior que), “<” (menor que), “>=” (maior ou igual a), “<=” (menor ou igual a).

Exemplo de código 6 – Condição (linguagem Fadas)

Sintaxe Condição:	Se comparação Então #lista de comandos Fim Se.
Exemplo:	Se Total = 1 Então Carro:Move “Para cima”;5;2. Fim Se.

Alternativamente, através da palavra reservada “Senão”, pode-se definir um bloco de comandos a serem executados caso a comparação seja falsa. A sintaxe desta alternativa é apresentada no exemplo de código 7.

Exemplo de código 7 – Comparação (linguagem Fadas)

Sintaxe Condição:	Se comparação Então #lista de comandos para comparação verdadeira Senão #lista de comandos para comparação falsa Fim Se.
-------------------	--

4.6.1.6 Comando em laço

O comando em laço ou *loop*, apresentado no exemplo de código 8, permite que o “usuário-autor” crie um grupo de comandos a serem executados repetidas vezes, de acordo com uma comparação. O comando de laço é iniciado pela palavra reservada “Enquanto” seguida de uma comparação numérica e da lista de comandos. Ele é fechado com a palavra reservada “Fim Enquanto” seguida do ponto final.

Exemplo de código 8 – Laço (linguagem Fadas)

Sintaxe Laço:	Enquanto comparação #lista de comandos para comparação verdadeira Fim Enquanto.
Exemplo:	Enquanto Contador < 10 Carro:Move “para frente”; 1; 2. Contador = Contador + 1. Fim Enquanto.

4.6.1.7 Rotina criada pelo usuário

Uma rotina é um conjunto de comandos agrupados criados pelo “usuário-autor”, e associados a um nome pelo qual a rotina será chamada para a execução do conjunto de comandos. A declaração de uma rotina, apresentada no exemplo de código 9, é iniciada com as palavras reservadas “Cria Rotina”, um identificador e opcionalmente uma lista de parâmetros. A seguir os comandos da rotina são definidos até que a palavra reservada “Fim Rotina” e o ponto final encerrem a declaração. Uma rotina não pode ser declarada mais do que uma vez.

Exemplo de código 9 – Declaração de Rotina (linguagem Fadas)

Sintaxe Declaração Rotina:	Cria Rotina NomeRotina [Parâmetros] #lista de comandos Fim Rotina.
Exemplo:	Cria Rotina Soma Num1; Num2; Resultado Resultado = Num1 + Num2. Fim Rotina.

Dentro de uma rotina, também é permitido a declaração de caixas, sendo que estas só podem ser usadas pelos comandos locais também pertencentes a rotina.

A chamada de uma rotina para a execução de seus comandos só é válida se a rotina foi declarada previamente. A declaração da rotina, apresentada no exemplo de código 10, é iniciada pelo identificador da rotina, a seguir os parâmetros definidos na declaração da mesma e o ponto final.

Exemplo de código 10 – Chamada de Rotina (linguagem Fadas)

Sintaxe Chamada Rotina:	NomeRotina [Parâmetros].
Exemplo:	Soma 1000; 2000; Resultado.

4.6.1.7 Comentários

O “usuário-autor” pode inserir comentários no código do programa através do símbolo “#”. Todos os demais caracteres encontrados após este símbolo até o final da linha serão desprezados pelo interpretador. Os comentários são altamente recomendados para aumentar a legibilidade do programa.

4.6.1.8 Eventos

Um recurso importante para a interação com o mundo virtual é a programação de eventos. A linguagem Fadas permite ao “usuário-autor” criar rotinas a serem executadas quando eventos pré-definidos, associados a um objeto, ocorrerem. Exemplos de eventos são o clique do *mouse*, o pressionamento de uma tecla do teclado, o pressionamento de um botão do joystick e a colisão de um objeto com outro.

Primeiramente o evento tem que ser declarado, conforme o exemplo de código 11, com a palavra reservada “Cria Evento” seguida de um identificador da rotina a ser executada, um texto contendo o tipo do evento, o símbolo “;”, um texto contendo o nome do objeto que detecta o evento e o ponto final. Somente um evento de um mesmo tipo por objeto, pode ser declarado. Por exemplo, não é permitido criar duas rotinas de tratamento do evento do tipo “Botão Joystick Pressionado” para o objeto de nome “Mundo”.

Exemplo de código 11 – Declaração de Evento (linguagem Fadas)

Sintaxe declaração evento:Cria Evento NomeRotina; TextoTipoEvento; TextoNomeObjeto.	
Exemplos	Cria Evento TrataCliqueMouse; “Clique Mouse”; “Mundo”. Criar Evento TrataColisaoCarro; “Colisão”; “Carro”.

4.6.2 Exemplos

A seguir apresentam-se alguns exemplos de código na linguagem Fadas.

4.6.2.1 Movimentação

O exemplo de código 12 apresenta comandos de movimentação enquanto o exemplo de código 13 efetua os mesmos movimentos com caixas e uma estrutura de repetição.

Exemplo de código 12 – Movimento 1 (linguagem Fadas)

<pre># movendo-se o objeto Pretinha na trajetória de um quadrado de lado 10 # com duração de 2 segundos Programa Pretinha:Move “para frente”;10;2. Pretinha:Vira “para direita”. Pretinha:Move “para frente”;10;2. Pretinha:Vira “para direita”. Pretinha:Move “para frente; 10;2. Pretinha:Vira “para direita”. Pretinha:Move “para frente”; 10;2. Pretinha:Vira “para direita”. # fecha o mundo atual Mundo:Fecha. Fim Programa.</pre>
--

Exemplo de código 13 – Movimento 2 (linguagem Fadas)

```
# movimento utilizando-se caixas para guardar valores
# e estrutura de repetição.
Programa
  Cria Caixa LadoQuadrado = 10.
  Cria Caixa AnguloQuadrado = 90.
  Cria Caixa TotalLados = 4.
  Cria Caixa Contador = 1.

  Enquanto Contador <= TotalLados
    Pretinha:Move “para frente”; LadoQuadrado.
    Pretinha:Vira “para direita”; AnguloQuadrado.
    Contador = Contador + 1.
  Fim Enquanto.

# fecha o mundo atual
Mundo:Fecha.

Fim Programa.
```

4.6.2.2 Rotina criada pelo usuário

O exemplo de código 14 , escrito na linguagem Fadas, apresenta a declaração e chamada de uma rotina utilizada para a conversão de temperaturas.

Exemplo de código 14 – Rotina (linguagem Fadas)

```
#o programa finalizará quando o usuário fechar a janela do mundo virtual
Programa
  # rotina para conversão de temperatura em Celcius para Fahrenheit
  Cria Rotina ConverteFahrenheit TemperaturaCelcius; Resultado
  Resultado = (TemperaturaCelcius * 9 / 5 ) + 32.
  Fim Rotina

#converte a temperatura de 30 graus Celcius em Fahrenheit
Cria Caixa TempF.
ConverteFahrenheit 30; TempF.

#cria uma caixa de texto 3D com o nome txtTempF
# na posição atual do objeto Branquinha
# com o texto da caixa TempF
Branquinha:CriaCaixaTexto3D TempF; txtTempF.

Fim Programa.
```

4.6.2.3 Codificando Eventos

O exemplo de código 15, escrito na linguagem Fadas, trata o evento do tipo “Tecla Pressionada” para processar comandos de acordo com a tecla pressionada pelo usuário.

Exemplo de código 15 – Evento (linguagem Fadas)

```

Programa
# declara evento de tecla pressionada
Cria Evento TrataTeclas “Tecla Pressionada”; “Mundo”.
# declaração de caixas com valores pré-definidos para teclas
Cria Caixa TECLA_ESCAPE = 1.
Cria Caixa TECLA_A = 30.
Cria Caixa TECLA_D = 32.
Cria Caixa TECLA_E = 18.
Cria Caixa TECLA_S = 31.
Cria Caixa TECLA_X = 45.
Cria Caixa TECLA_F2= 60.
# declara rotina para tratar evento do tipo “Tecla Pressionada”
Cria Rotina TrataTeclas CodigoTecla
  Se CodigoTecla = TECLA_ESCAPE Então
    #encerra o programa se a tecla escape for pressionada
    Mundo:Fecha.
  Fim Se.
  Se CodigoTecla = TECLA_D Então
    # se a tecla D for pressionada
    #vira o objeto Pretinha 30 graus para a esquerda em 3 segundos
    Pretinha:Vira “para esquerda”;30;3.
  Fim Se.
  Se CodigoTecla = TECLA_S Então
    # se a tecla S for pressionada
    #vira o objeto Pretinha 30 graus para a direita em 3 segundos
    Pretinha:Vira “para direita”;30;3.
  Fim Se.
  Se CodigoTecla = TECLA_E Então
    # se a tecla E for pressionada
    # soma um na velocidade do objeto Pretinha.
    Pretinha:Velocidade = Pretinha:Velocidade + 1.
  Fim Se.
  Se CodigoTecla = TECLA_X Então
    # se a tecla X for pressionada
    # subtrai um na velocidade do objeto Pretinha.
    Pretinha:Velocidade = Pretinha:Velocidade - 1.
  Fim Se.
  Se CodigoTecla = TECLA_F2 Então
    # se a tecla F2 for pressionada
    # esconde o objeto Pretinha
    Pretinha:Esconde.
  Fim Se.
Fim Rotina
Fim Programa.

```

4.7 Interpretador

4.7.1 Reconhecedor Léxico Sintático

O interpretador do sistema FADAS é baseado em reconhecedores da linguagem. Um reconhecedor, através de testes, classifica um texto como válido ou não para a linguagem. A partir da descrição formal da gramática na notação de Wirth, foram criados os reconhecedores. Este mapeamento da gramática em reconhecedores foi feito através da técnica descrita pelo pesquisador José Neto (1987) implementada em um programa gerador de reconhecedor léxico sintático, cuja tela é apresentada na figura 26.

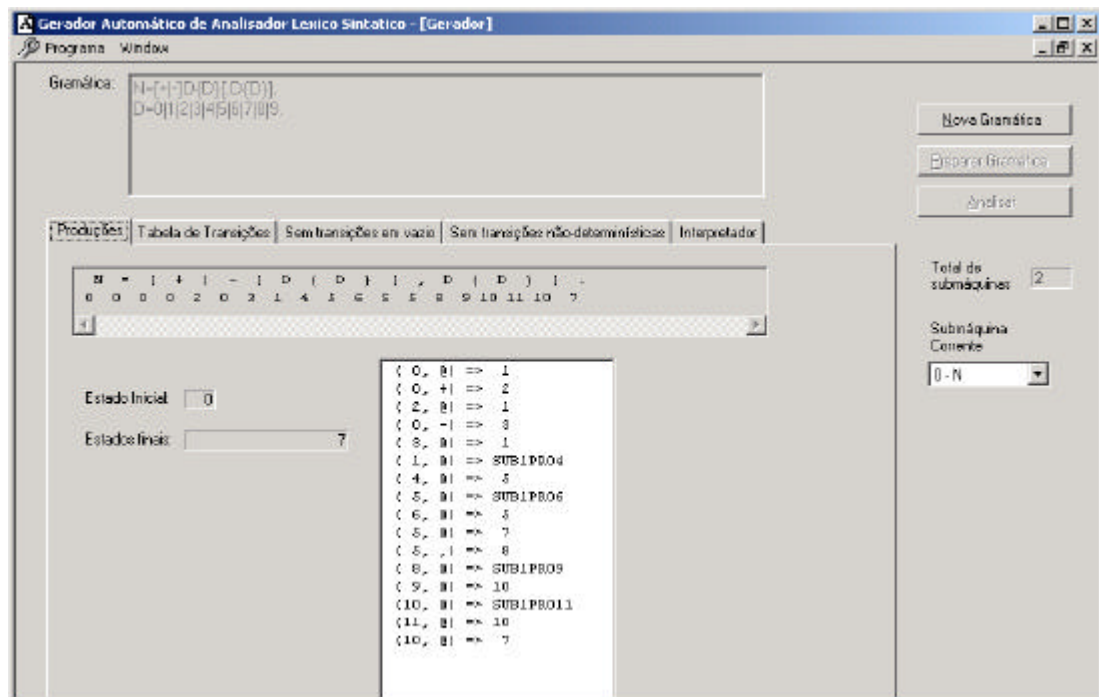


Fig. 26 – Gerador de Reconhecedor Léxico Sintático

A classe CAutomatoPilha, cuja declaração implementada em linguagem C++ é apresentada no exemplo de código 16, é a responsável por reconhecer um texto proposto para a linguagem Fadas.

Exemplo de código 16 – Reconhecedor (linguagem C++)

```

class CAutomatoPilha
{
// atributos públicos
public:
    CColecao Submaquinas;
// atributos privados
private:
    CTexto msCadeiaEntrada;
    CColecao Pilha; // coleção de elementos de pilha
    CColecao Estagios; // estagios salvos para tratar indeterminismo
// métodos públicos
public:
    CAutomatoPilha(void);
    ~CAutomatoPilha(void);
    void AdicionaSubmaquina(CSubmaquina *pSubmaquina);
    void DefineCadeiaEntrada( CTexto sCadeiaEntrada);
    bool Reconhece(void);
    bool Reconhece(CTexto &sCadeiaEntrada);
// métodos privados
private:
    void Empilhar(int iSubmaquinaRetorno, int iEstadoRetorno);
    void Desempilhar(int &iSubmaquinaRetorno, int &iEstadoRetorno);
};

```

4.7.2 Analisadores Léxico, Sintático e Semântico

O interpretador do sistema FADAS, desenvolvido em linguagem C++, tem a sua funcionalidade encapsulada na classe CInterpretador cuja declaração é apresentada no exemplo de código 17. Esta classe é composta de objetos de três outras classes, responsáveis pelas principais funções do interpretador: CAnalisadorLexico, CAnalisadorSintatico e CAnalisadorSemantico. Os identificadores e palavras reservadas da linguagem, são armazenadas em coleções da classe CTexto que encapsula as cadeias de caracteres.

Exemplo de código 17 – Interpretador (linguagem C++)

```

class CInterpretador
{
// atributos públicos
public:
    CAnalizadorLexico Lexico;
    CAnalizadorSintatico Sintatico;
    CAnalizadorSemantico Semantico;
    CColecao Identificadores;           // lista (CTexto) de identificadores
    CColecao PalavrasReservadas;       // lista (CTexto) de palavras reservadas
    CColecao Linhas;                   // lista (CLinhaCodigo) de linhas do
// programa
// atributos privados
private:
    LONGO_SEM_SINAL ulLinhaAtual;
// construtores
public:
    CInterpretador();
    ~CInterpretador(void);
// métodos públicos
public:
    bool ProcessaCodigo(CTexto &sCodigo, int &iLinhaErro);
// métodos privados
private:
    void MontaPalavrasReservadas(void);
};

```

O analisador léxico do interpretador recebe as cadeias de caracteres tal como digitadas pelo usuário e elimina espaços em branco, caracteres de mudança de linha e comentários, através de um filtro no início da análise léxica. A seguir o texto é classificado e convertido em átomos. Um átomo é o menor elemento com significado para o interpretador Fadas. Os tipos de átomos implementados são: palavra reservada, identificador, número, texto, cor, sinal e vetor. A classificação é feita por reconhecedores sendo que a palavra reservada é inicialmente classificada como identificador e após uma pesquisa na coleção de palavras reservadas reclassificada.

O analisador sintático classifica os comandos da linguagem através de reconhecedores. O usuário pode solicitar a execução de um comando ou do programa. No caso de um comando, o analisador léxico entrega uma coleção de átomos para o analisador sintático classificar. Quando o usuário solicita a execução de um programa, o analisador léxico entrega para o analisador sintático, todas as linhas constituídas cada uma por uma coleção de átomos. Então o analisador sintático

as classifica verificando a sintaxe de todo o programa. Se ele encontrar um erro, a mensagem de erro será mostrada no janela de resposta indicando a linha com o erro.

A seguir o analisador semântico se encarrega de executar no ambiente Fadas, o significado de cada comando ou atribuição, como calcular expressões e comparações numéricas, instanciar classes dinamicamente para armazenar números, textos, vetores ou cores; executar métodos dos objetos tridimensionais do mundo virtual.

4.7.3 Tratamento de Erros

Cada um dos três analisadores: léxico, sintático e semântico; ao detectar um erro salva as informações como código e mensagem original, fonte e mensagem para o usuário. A mensagem de erro original, proveniente de alguma biblioteca é separada de uma mensagem mais amigável, preparada pelo interpretador, como por exemplo “caractere inválido”, “comando não reconhecido”, “direção inválida”, “objeto não encontrado”. A lista de erros implementada no sistema se encontra no apêndice IV desta dissertação.

O usuário pode configurar o sistema para gravar seqüencialmente cada erro em um arquivo texto de *log* com as informações de data, hora, código original, mensagem original e mensagem do usuário.

Outra opção do usuário em relação ao tratamento de erros, é configurar a resposta sonora. Quando ela está ativada, o sistema FADAS reproduz o arquivo de voz da mensagem de erro, sempre que um erro é encontrado.

4.8 Biblioteca Gráfica OpenGL

A aplicação Mundo Virtual utiliza a biblioteca OpenGL para criar os dados gráficos e sintetizar imagens 2D projetadas. Esta biblioteca é independente do hardware e é implementada em diversas plataformas de *hardware* e *software*. Aceleradores gráficos que implementam funções gráficas em seu *hardware* fornecem *drivers* OpenGL de maneira que o programador ao utilizar esta biblioteca está utilizando suas funcionalidades.

OpenGL foi desenvolvida originalmente pela Silicon Graphics, Inc. (SGI) como uma API gráfica multi-plataforma de múltiplos propósitos. Desde 1992, a

especificação deste padrão tem sido de responsabilidade da OpenGL Architecture Review Board (ARB), formada pelos principais fornecedores gráficos (Hawkins; Astle, 2001).

Cada objeto 3D do mundo virtual é formado por uma rede de triângulos. Quanto maior o número de triângulos, maior a qualidade da cena sintetizada e maior o processamento computacional requerido. Quando o usuário importa um modelo 3D o sistema FADAS carrega a rede de triângulos armazenada no arquivo do modelo de acordo com o seu formato. Para exibir o modelo 3D, o sistema varre todos os triângulos do modelo que estão armazenados em memória e a biblioteca OpenGL sintetiza a cena. O sistema FADAS suporta a importação de modelos 3D nos formatos de arquivo MD2, o modelo do jogo Quake, e o formato de arquivo ASE, um formato ASCII exportado pelo modelador 3D Studio Max.

A cena, no sistema FADAS, é projetada em perspectiva, de acordo com a posição da câmera ativa e o volume de visualização apresentado na figura 27, que é uma pirâmide cortada no topo (*frustum*). Os objetos localizados fora deste volume, especificado em OpenGL pela função `glFrustum`, não são sintetizados na cena. A projeção em perspectiva, segundo Foley et al. (1990), gera realismo para a cena, sendo similar aos sistema fotográfico e sistema visual humano: quanto maior a distância do objeto em relação a câmera, menor ele aparece na imagem final.

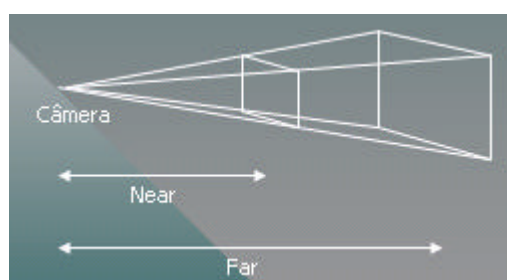


Fig. 27 – Volume de Visualização

Um dos principais fatores do processo de síntese de cenas 3D e que contribui fortemente com o realismo da imagem final é a iluminação. No modelo de luz da biblioteca OpenGL, a cor da fonte de luz é definida pela quantidade de vermelho, verde e azul que emitem sobre as superfícies dos materiais dos objetos que compõem a cena, e são caracterizadas pelo percentual das componentes vermelho,

verde e azul que refletem em várias direções. Algumas luzes vêm de uma direção ou de uma posição no espaço e outras são espalhadas por todo o ambiente.

O modelo de iluminação OpenGL considera que a iluminação é dividida em quatro componentes calculados independentemente e adicionados em conjunto. A luz ambiente parece vir de todas as direções e também é espalhada para todas as direções quando encontra uma superfície. A luz difusa vem de uma direção e quando encontra uma superfície também é espalhada por todas as direções. A luz especular vem de uma direção e é refletida de maneira organizada como em metais brilhantes. A luz emissiva (*emissive*) é utilizada para que a luz pareça estar sendo emitida por um objeto e não afeta nenhum outro objeto da cena.

No sistema FADAS, o usuário pode definir até oito fontes de luz, definindo as componentes de luz ambiente, difusa e especular, a posição no espaço ou a sua direção. O usuário define um nome para cada fonte de luz e pode ligá-la ou desligá-la através de comandos de ação. Ao criar um mundo virtual, o sistema FADAS cria automaticamente um fonte de luz branca difusa e espectral com a intensidade máxima, na direção do usuário para a tela, ou seja, entrando na tela do computador. Esta luz recebe o nome de “Principal”. Uma pequena luz branca ambiente também é definida de modo que os objetos serão vistos mesmo que nenhuma fonte de luz seja definida. Esta luz recebe o nome de “Global”. A janela “Fontes de Luz”, apresentada na figura 25, permite que o usuário visualize e configure as fontes de luz da cena.

No sistema FADAS, assim como no mundo real, quando se especifica uma cor para um objeto, se está definindo na realidade as componentes de luz vermelha, verde e azul das luzes ambiente, difusa, e especular que o objeto reflete.

Se o usuário desejar definir a luz refletida por cada componente de luz ele o fará como no exemplo de código 18.

Exemplo de código 18 – Cor (linguagem Fadas)

Carro:Cor = C(0;0;100).	#carro de cor azul
Carro:CorLuzAmbiente = C(0;0;100).	#carro reflete luz azul ambiente
Carro:CorLuzDifusa = C(0;0;100).	#carro reflete luz azul difusa
Carro:CorLuzEspecular = C(0;0;100).	#carro reflete luz azul especular

O usuário pode simular um objeto que emite luz, definindo a luz emissiva do objeto como no exemplo de código 19.

Exemplo de código 19 – Luz Emissiva (linguagem Fadas)

Lampada:LuzEmissiva = C(100;0;0).	#lâmpada emite luz vermelha
-----------------------------------	-----------------------------

Outro recurso da biblioteca OpenGL utilizado para o sistema FADAS é definir a opacidade de um objeto. No exemplo de código 20, o objeto se torna semi-transparente.

Exemplo de código 20 – Opacidade (linguagem Fadas)

Branquinha:Opacidade = 50.	#branquinha com 50% de opacidade
----------------------------	----------------------------------

Quanto maior o número de fontes de luz, mais recursos computacionais serão exigidos nos cálculos. Para iluminar os modelos 3D formados por redes de triângulos, o sistema FADAS fornece duas opções: calcular a normal de cada triângulo ou calcular a normal de cada vértice. A classe CVetor encapsula um vetor em algumas de suas operações como adição, subtração, multiplicação e divisão por um escalar, produto vetorial, produto escalar e normalização. Logicamente o recurso computacional gasto para gerar as normais e iluminar um objeto com normais em cada vértice é maior, porém a qualidade da imagem final é sensivelmente melhor.

O recurso gráfico que traz mais realismo a um modelo 3D é o mapeamento de texturas. Através dele imagens são acopladas aos polígonos como uma pele. As ações que o usuário aplica ao objeto são também aplicadas à textura igualmente. É possível utilizar qualquer imagem, como por exemplo fotografias tiradas do mundo real, trazendo assim maior realismo.

O sistema FADAS utiliza texturas 2D carregadas a partir de arquivos de imagem nos formatos BMP de 24 bits, PCX e PPM. Sendo que um objeto no sistema FADAS pode ter várias texturas carregadas e uma ativa.

O modelos 3D geralmente utilizam textura e fornecem as coordenadas de textura para cada vértice. O sistema FADAS, além de carregar as imagens, carrega as coordenadas de textura e as informa para a biblioteca OpenGL com cada vértice da rede de triângulos do modelo. Tanto o formato ASE quanto o formato MD2 de modelos 3D, cuja importação é suportada pelo sistema FADAS, podem conter texturas a serem utilizadas no sistema FADAS.

Outro recurso de textura utilizado pelo sistema FADAS é o uso e construção automática de *mipmaps* através da função `gluBuild2DMipmap`. Quando a câmera se aproxima ou se afasta de um objeto com textura, a biblioteca OpenGL tenta compensar a textura para que faça um mapeamento adequado. O uso de *mipmaps* corresponde a gerar imagens de vários tamanhos da mesma textura para ser mapeada adequadamente a um objeto perto da câmera ou afastado.

Os textos tridimensionais que o “usuário-autor” pode criar e manipular no sistema FADAS também são sintetizados pela biblioteca OpenGL.

4.9 Imagens Estereoscópicas

A biblioteca OpenGL fornece interface para a projeção de imagens estereoscópicas desde que a placa controladora de vídeo suporte este recurso. Se houver este suporte e o usuário desejar ativá-lo, o sistema FADAS sintetiza a cena em duas câmeras levemente deslocadas, simulando os olhos direito e esquerdo do usuário. Akka (1998) descreve como gerar imagens estereoscópicas com a biblioteca OpenGL.

A geração de imagens estereoscópicas pode ser ativada tanto em ambiente de uma única projeção como o de um computador pessoal, quanto no ambiente de múltiplas projeções da CAVERNA Digital. Akka (1998) recomenda o balanceamento do efeito do paralaxe, de modo que exista tanto paralaxe negativo quando positivo, o que pode ser feito colocando-se o centro de interesse no paralaxe zero. Conforme recomendações (StereoGraphics, 1997), o paralaxe de uma imagem estereoscópica deve ser o menor possível mantendo o efeito de profundidade. A

recomendação é de não ultrapassar o ângulo de 1,5 graus, assim, em um computador pessoal, o usuário está geralmente a 50 cm do monitor, e portanto o paralaxe não deve ser maior que 13 mm. Na CAVERNA Digital, o usuário quando está no centro tem uma distância de 1,5 metros para o plano de projeção, e portanto o paralaxe entre as imagens não deve ser maior do que 39,3 mm. Porém estas regras podem ser quebradas pois o desconforto e o efeito estereoscópico varia de pessoa para pessoa.

O sistema FADAS, utiliza o modelo de câmeras com lentes paralelas apresentado na figura 28 pela StereoGraphics. A distância t_c entre as câmeras pode ser regulada pelo usuário assim como a translação horizontal da imagem (HIT), apresentado na figura 29, que serve para controlar o paralaxe. Alterando-se o HIT, altera-se a distância para o plano de paralaxe zero (ZPS).

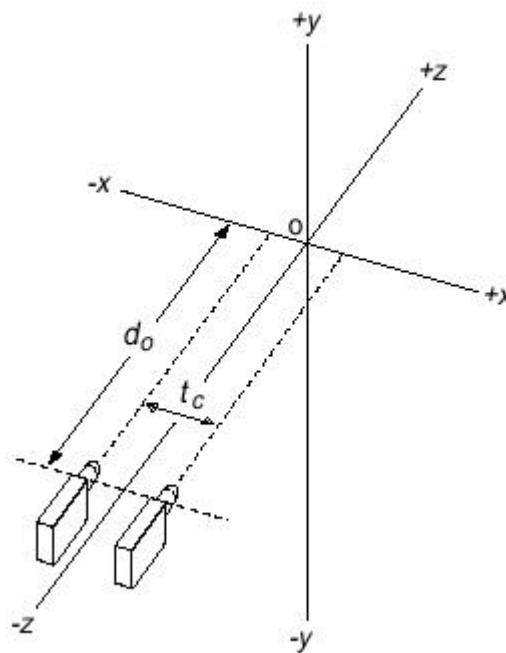


Fig. 28 – Modelo de câmeras com eixos paralelos (STEREOGRAPHICS, 1997)

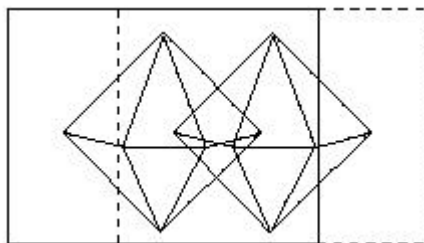


Fig. 29 – Translação horizontal da imagem (HIT) (STEREOGRAPHICS, 1997)

Os seguintes comandos foram disponibilizados pelo sistema FADAS para o usuário controlar a estereoscopia: Config:AumentaDistanciaCameras, Config:DiminuiDistanciaCameras, Config:AumentaHIT, Config:DiminuiHIT.

4.10 Movimento e Rotação Relativa

O sistema de coordenadas global do mundo virtual no sistema FADAS, apresentado na figura 30, segue o padrão da biblioteca OpenGL. O sistema mantém a posição central de cada objeto, assim como um vetor de deslocamento. A cada quadro, o sistema efetua a translação do objeto para a sua posição atual.

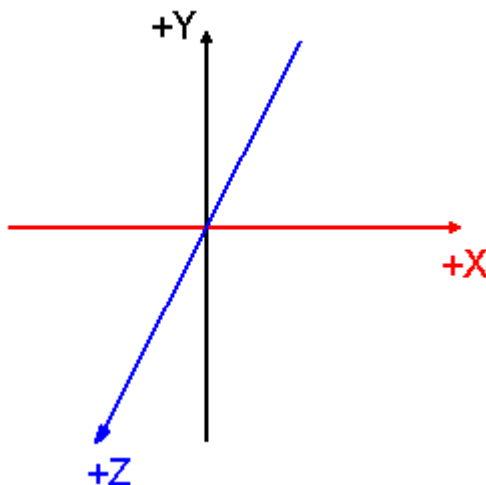


Fig. 30 - Sistema de coordenadas global

Quando um objeto é adicionado ao mundo virtual, ele é colocado na posição central (coordenadas globais x, y e z zeradas) com o seu eixo Z, que representa a direção frontal ou “para frente”, alinhado com o eixo Z global. Se já houver algum

objeto neste local, o novo objeto é deslocado para a direita ou esquerda até encontrar um espaço livre. A verificação de colisão entre objetos está descrita no item 5.13.

Opcionalmente um objeto pode ser adicionado ao mundo virtual na mesma posição e orientação de um objeto existente.

Os recursos de movimentação e rotação relativa fornecem ao usuário grande flexibilidade de movimentação no espaço. O sistema FADAS mantém para cada objeto tridimensional um sistema de coordenadas locais, assim, quando o usuário solicita uma movimentação para frente, o sistema efetua o deslocamento de acordo com o eixo Z local do objeto. A direção “para esquerda” e a direção “para cima” são representadas pelos eixos locais X e Y respectivamente.

A rotação relativa implementada no sistema FADAS, permite ao usuário solicitar uma rotação anti-horária ou horária de um objeto em relação ao seu eixo Y local (virar “para esquerda” ou “para direita”), solicitar uma rotação anti-horária ou horária em relação ao seu eixo X local (inclinar “para baixo” ou “para cima”), e permite solicitar uma rotação anti-horária ou horária ou em relação ao seu eixo Z local (angular “para direita” ou “para esquerda”).

A implementação da rotação relativa e manutenção das coordenadas locais dos objetos tridimensionais, foi baseada na matriz R de rotação ortogonal descrita por Gruber (2000) cujas propriedades foram fornecidas por Pique apud Gruber (2000) e são descritas abaixo.

- a) R é normalizada: o quadrado dos elementos em qualquer linha ou coluna soma um;
- b) R é ortogonal: o produto escalar de qualquer par de linhas ou qualquer par de colunas é zero;
- c) As linhas de R representam as coordenadas no espaço original de unidades de vetor ao longo dos eixos coordenados do espaço que sofre rotação;
- d) As colunas de R representam as coordenadas no espaço que sofreu rotação de unidades de vetor ao longo dos eixos do espaço original.

O sistema FADAS armazena para cada objeto 3D uma matriz de rotação implementada através da classe CMatriz.

4.11 Navegação

No protótipo do sistema FADAS foram criadas interfaces para navegação no mundo virtual através de teclado e *joystick*. Os comandos destas interfaces são apresentados nas tabelas II e III respectivamente. Quando o usuário aciona uma tecla o sistema executa um comando de deslocamento de uma distância padrão ou rotação de um ângulo padrão. Sendo que tanto o deslocamento padrão quanto a rotação padrão podem ser aumentadas ou diminuídas através de comando enviados à câmera ativa ou pela teclas “+” e “-”. Desta maneira o usuário pode adequar a navegação de acordo com a agilidade desejada e tamanho do ambiente virtual.

Para controlar os dispositivos de entrada (*mouse*, teclado e *joystick*), ao invés de utilizar os mecanismos tradicionais como a interface Win32 e mensagens do sistema operacional Microsoft® Windows, o sistema FADAS utiliza a interface de programação Microsoft® DirectInput, porque esta interface se comunica diretamente com os *device drivers*² que controlam os dispositivos, fornecendo respostas mais rápidas além de incluir suporte a recursos especiais como resposta à pressão (*force feedback*).

Tabela II – Navegação com teclas

Tecla	Comando
Seta para baixo	Move deslocamento para trás
Seta para cima	Move deslocamento para frente
Seta para direita	Move deslocamento para direita
Seta para esquerda	Move deslocamento para esquerda
S	Vira ângulo para esquerda
D	Vira ângulo para direita
X	Inclina ângulo para baixo
E	Inclina ângulo para cima
PGDN	Move deslocamento para baixo
PGUP	Move deslocamento para cima

² Device driver é o código de programa que controla um dispositivo. Geralmente é fornecido pelo fabricante do dispositivo.

Tabela III – Navegação com *joystick*

Botão ou deslocamento no eixo	Comando
EixoY – Positivo	Move deslocamento para trás
EixoY – Negativo	Move deslocamento para frente
EixoX – Positivo	Move deslocamento para direita
EixoX – Negativo	Move deslocamento para esquerda
Botão 0	Vira ângulo para esquerda
Botão 1	Vira ângulo para direita
Botão 2	Inclina ângulo para baixo
Botão 3	Inclina ângulo para cima
Botão 4	Move deslocamento para baixo
Botão 5	Move deslocamento para cima

Outro recurso que pode ser utilizado para navegar é atribuir uma velocidade para a câmera ativa na direção “para frente”. Assim o usuário “passeia” pelo mundo virtual podendo ou não alterar a direção da velocidade ao efetuar uma rotação. O comando `Câmera:RotacaoAfetaVelocidade` define se a direção da velocidade da câmera deve ser alterada com uma rotação ou se manter constante.

4.12 Animação

4.12.1 Animação baseada em script

No sistema FADAS o “usuário-autor” trabalha com animação interativa em tempo real. Através de comandos de movimento, rotação e deformação de objeto, o “usuário-autor” define a duração em segundos da ação, gerando assim animações nos objetos do mundo virtual.

Os movimentos são simulados pelo sistema FADAS através das leis de física mecânica. A fórmula form.(1) da função da velocidade e a fórmula form.(2) do movimento uniformemente variado, foram implementadas.

$$V = V_0 + a \cdot t \quad (1)$$

$$S = S_0 + V_0 \cdot t + \frac{1}{2} \cdot a \cdot t^2 \quad (2)$$

O usuário autor, para movimentar um objeto, pode diretamente definir a velocidade e aceleração de um objeto, ou simplesmente definir um deslocamento a ser executado em determinada duração, para que o sistema FADAS calcule a

velocidade conseqüentemente e a posição do objeto a cada quadro. Para efetuar os cálculos, o sistema FADAS guarda o intervalo de tempo decorrido entre cada quadro da cena sintetizada e projetada.

No exemplo de código 21, escrito na linguagem Fadas, o texto tridimensional de nome “txtOla” e o objeto de nome “Boneco” sofrem simultaneamente uma rotação de 360 graus em dez segundos.

Exemplo de código 21 – Texto tridimensional (linguagem Fadas)

```
txtOla:Vira "para esquerda";360;10.
Boneco:Vira "para esquerda";360;10.
```

No exemplo de código 22, escrito na linguagem Fadas, o objeto de nome “Branquinha” é deslocado 20 unidades para frente em 10 segundos. Após esta animação, ele faz uma rotação de 90 graus para a direita em 3 segundos e se desloca a seguir novamente 20 unidades para frente em 10 segundos.

Exemplo de código 22 - Animação baseada em script (linguagem Fadas)

```
Branquinha:Move "para frente";20;10.
Mundo:Espera 10.
Branquinha:Vira "para direita";90;3.
Mundo:Espera 3.
Branquinha:Move "para frente";20;10.
```

4.12.2 Animação baseada em keyframe

O sistema FADAS também implementa animação em tempo real através da técnica de keyframe. Neste caso é necessário que o modelo 3D importado pelo sistema FADAS, possua conjuntos de frames ou quadros chaves preparados pelo autor do modelo. Cada conjunto de quadros recebe um nome pelo qual o “usuário-autor” pode invocar a execução.

Entre um quadro chave e o próximo quadro chave, o sistema FADAS calcula quadros intermediários através de interpolação linear. O número de quadros intermediários depende do tempo médio gasto para sintetizar e projetar um quadro no computador onde o sistema está instalado e do tempo total em que o ciclo da animação deve durar. Por exemplo, em um computador com uma taxa média de 20

quadros por segundo, cada quadro custa 0,05 segundos. Para executar uma animação em 2 segundos, são necessários 40 quadros. Se a animação possui 6 quadros chaves, o sistema deve criar 34 quadros interpolados. A cada par de quadros é possível inserir 6 quadros interpolados, num total de 30 quadros interpolados entre o primeiro e o último dos 6 quadros chaves. Ao final restaram ainda dois décimos de segundo do tempo total que a animação deveria durar, mas este cálculo é aproximado já que o tempo real de cada quadro varia de ciclo para ciclo do *loop* principal* do sistema FADAS.

No exemplo de código 23, o primeiro comando na linguagem Fadas inicia um movimento animado baseado em script e no segundo comando, executa a animação com a técnica keyframe no mesmo objeto, combinado as duas técnicas de animação na mesma cena.

Exemplo de código 23 – Animação baseada em *keyframe* (linguagem Fadas)

```
# move o objeto Ogro a distância de 20 unidades durante 10 segundos.
Ogro:Move "pra frente";20;10.
# executa a animação Correndo, durante 10 segundos, com um ciclo de animação definido em 2
segundos.
Ogro:Anima "Correndo";10;2.
```

4.13 Detecção de Colisão entre Objetos

Para detectar colisão entre os objetos no mundo virtual, o sistema FADAS calcula uma caixa em volta de cada objeto 3D, varrendo todos os vértices de cada objeto e armazenando as coordenadas mínima e máxima em cada um dos eixos x, y e z. Desta forma as faces da caixa resultante são paralelas os eixos x, y e z. A cada movimento ou distorção do objeto, o sistema atualiza as coordenadas da caixa e verifica possíveis colisões.

O código para testar a colisão com outro objeto é simples e é apresentado no exemplo de código 24, escrito em linguagem C++.

* O ciclo principal implementado no sistema FADAS está descrito no item 4.14 - Ciclo Principal da Aplicação Mundo Virtual.

Exemplo de código 24 – Detecção de colisão (linguagem C++)

```

bool CObjeto3D::DetectaColisao(CObjeto3D *pObj)
{
    bool bRet(false);
    if ( (pObj->CaixaEmVolta.xMax >= CaixaEmVolta.xMin && pObj->CaixaEmVolta.xMin
<= CaixaEmVolta.xMax) &&
        (pObj->CaixaEmVolta.yMax >= CaixaEmVolta.yMin && pObj->CaixaEmVolta.yMin
<= CaixaEmVolta.yMax) &&
        (pObj->CaixaEmVolta.zMax >= CaixaEmVolta.zMin && pObj->CaixaEmVolta.zMin
<= CaixaEmVolta.zMax) )
        bRet = true;
    return bRet;
}

```

No protótipo do sistema FADAS apenas esta técnica simples foi implementada. Outros métodos de detecção de colisão podem ser implementados no futuro, como por exemplo dividir a caixa em volta do objeto em várias caixas menores para se obter maior precisão.

4.14 Ciclo Principal da Aplicação Mundo Virtual

O ciclo principal da aplicação Mundo Virtual que controla o mundo virtual é o seguinte:

- 1) Trata entrada de dados: teclado e joystick.
- 2) Salva tempo decorrido.
- 3) Calcula animações: movimentos, rotações e detecção de colisão.
- 4) Trata mensagens Windows: eventos gerados pela linguagem Fadas e comunicação Winsock.
- 5) Trata comandos recebidos na linguagem Fadas.
- 6) Desenha objetos tridimensionais do mundo virtual.

4.15 Arquivos do Mundo Virtual

Quando o usuário solicita que um mundo virtual seja salvo, o sistema FADAS gera um arquivo ASCII com a extensão “fap”. Neste arquivo, cada objeto do

mundo virtual é listado, incluindo também câmeras, fontes de luz e sons. As características de cada objeto em relação ao mundo virtual, tais como nome, posição, orientação, matriz de rotação e visibilidade, também são salvas neste arquivo.

Cada objeto também tem um arquivo dedicado para salvar as suas características próprias. Tanto estes arquivos de objeto quanto outros arquivos relacionados como de textura e de som, são salvos no mesmo diretório do arquivo principal do mundo virtual. O cabeçalho da estrutura do arquivo de objeto é mostrado no exemplo de código 25.

Exemplo de código 25 – Estrutura de arquivo de objeto 3D (linguagem C++)

```

struct ST_FADAS_CABECALHO_OBJETO
{
    char ID[4]; // identifica uma arquivo Fadas (FADA)
    int Versao; // versão do arquivo (1)
    char szNomeObjeto[TAMANHO_MAX_NOME]; // nome do objeto
    char szNomePai[TAMANHO_MAX_NOME]; // nome do objeto pai
    TIPO_CLASSE TipoClasse; // nome da classe do objeto
    TIPO_MODELO TipoModeloOriginal; // tipo de modelo original (ASE, MD2)
    int NumVertices; // número de vértices de um frame
    int NumCorTex; // número de coordenadas de textura
    int NumTriangulos; // número de triângulos
    int NumFrames; // número de Frames
    int NumAnimacoes; // número de animações
    int NumTexturas; // número de texturas
    ST_FADAS_MATERIAL Material; // material do objeto
    int iReservado1;
    int iReservado2;
    int iReservado3;
    long lReservado1;
    long lReservado2;
    long lReservado3;
    float fReservado1;
    float fReservado2;
    float fReservado3;
    char szReservado1[TAMANHO_MAX_NOME];
    char szReservado2[TAMANHO_MAX_NOME];
    char szReservado3[TAMANHO_MAX_NOME];
    fpos_t OffsetTriangulos; // Offset Triângulos
    fpos_t OffsetVertices; // Offset vértices
    fpos_t OffsetCortex; // Offset para coordenadas de textura
    fpos_t OffsetTexturas;
    fpos_t OffsetAnimacoes;
    fpos_t OffsetFim; // final do objeto
};

```

4.16 Som

O sistema FADAS utiliza a interface de programação Microsoft® DirectX Audio para reproduzir sons digitalizados ou sintetizados. Esta interface permite obter maior performance e controle dos dispositivos de audio, reproduzindo segmentos de som simultaneamente além de suportar som tridimensional.

O “usuário-autor” deve definir o nome e localização de um arquivo de som nos formatos WAVE e MIDI assim como associar um nome ao som para adicioná-lo e manipulá-lo no mundo virtual, como no exemplo de código 26.

Exemplo de código 26 – Som (linguagem Fadas)

```
Mundo:AdicionaSom "C:\Fadas\Biblioteca\Som\sabia.mid"; "somSabia".
somSabia:Toca.
```

Um som tridimensional é adicionado ao mundo virtual por uma das Fadas do mundo virtual, pois o sistema utiliza a posição e orientação do objeto de nome Pretinha ou Branquinha para posicionar o som no espaço virtual. Um som tridimensional tem posição no espaço e pode ter velocidade e orientação.

As distâncias mínimas e máximas também podem ser definidas pelo usuário. A distância mínima, cujo valor padrão é 1, é aquela em que o volume do som está com a intensidade máxima. Quando o “usuário-explorador”, ou melhor a câmera ativa, está distante o dobro da distância mínima, o volume do som é reduzido a metade, e assim por diante. A distância máxima define a distância em que o som não mais será reproduzido. No exemplo de código 27, um som tridimensional é criado, configurado e reproduzido.

Exemplo de código 27 – Som 3D (linguagem Fadas)

```
Branquinha:AdicionaSom3D "C:\Fadas\Biblioteca\Som\explosao.wav"; "somExplosao".
somExplosao:DistanciaMinima = 5.
Camera:Move "para trás";20;10.
somExplosao:Toca.
```

4.17 Comunicação entre processos

A comunicação entre processos no sistema FADAS é feita através de classes que encapsulam a API Windows Sockets. Esta API define uma especificação para interface de rede de computadores. Ela foi baseada na implementação sockets BSD (*Berkeley Software Distribution*) 4.3 para o sistema operacional Unix[®] da Universidade da Califórnia em Berkeley, possuindo algumas extensões para a plataforma Windows.

Através da API Windows Sockets e o usando o protocolo de rede TCP/IP, os aplicativos do sistema FADAS conversam um com o outro, estando instalados em um mesmo computador ou em computadores diferentes interligados por uma rede local.

Como os aplicativos do sistema FADAS são executados em um mesmo computador ou em computadores interligados por uma rede local, adotou-se o serviço de datagramas UDP (*User Datagram Protocol*) da família TCP/IP (*Transmission Control Protocol/Internet Protocol*), que envia os pacotes de informação sem a necessidade de manter conexão.

4.18 Multi-projeção e Rastreamento do usuário

O sistema FADAS, para suportar a utilização na CAVERNA Digital com cinco planos distintos, define um processo responsável pela geração de imagem projetada. A aplicação mestre é a responsável pela projeção frontal da cena e por enviar os comandos, sempre que uma alteração ocorre no mundo virtual, para as quatro aplicações clientes, responsáveis pelas projeções das vistas da direita, esquerda, de trás e de baixo. A figura 31 apresenta três projeções sincronizadas de um mundo virtual.

Quando uma aplicação cliente, por exemplo a da vista da direita, recebe um comando da aplicação mestre, ela executa o comando e efetua uma rotação na câmera ativa, de noventa graus para a direita. A seguir ela envia uma resposta para a aplicação mestre, que é a responsável pelo sincronismo das projeções.

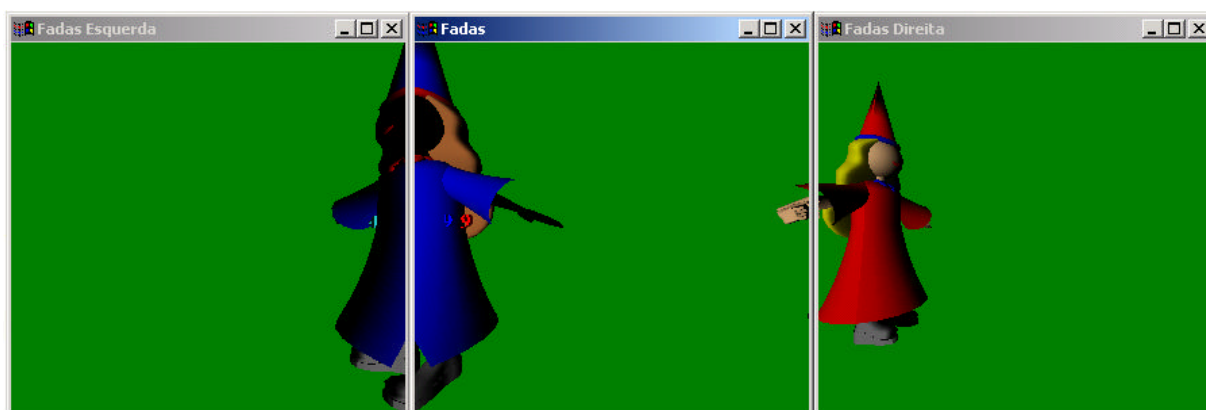


Fig. 31 – Duas projeções sincronizadas do sistema FADAS

O campo de visão horizontal do observador, é outro fator importante para o sistema multi-projeção, pois para o usuário sentir a imersão no ambiente de múltiplas projeções da CAVERNA Digital, as extremidades entre as projeções devem estar sincronizadas. Nesta primeira versão do sistema fadas, adotou-se um ângulo de noventa graus para o campo de visão do observador, que está visualizando o mundo sempre no centro do quadrado formado pelos quatro planos de projeção lateral, com uma distância fixa entre ele e os planos. Quando o observador navega pelo mundo virtual, o ângulo de visão e a distância aos planos são preservados, conforme a figura 32 que apresenta uma vista superior da CAVERNA Digital, onde o observador é representado pela cruz em azul e os campos de visão horizontal para cada um dos planos é representado pelo ângulo α .

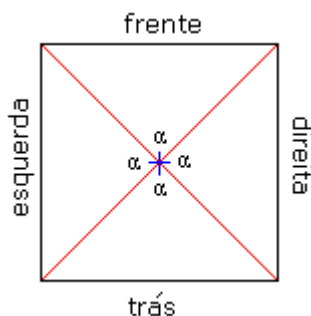


Fig. 32 – Observador no centro da CAVERNA Digital

Planeja-se implementar o rastreamento da localização do usuário dentro da CAVERNA Digital, ajustando-se a distância dele até os quatro planos de projeção e

conseqüentemente o campo de visão horizontal. No exemplo da figura 33a, o observador deu alguns passos para frente, reduzindo a distância até o plano de projeção frontal e aumentando o ângulo de visão α . Já no exemplo da figura 33b, o observador moveu-se de maneira que cada projeção tem seu próprio ângulo de visão.

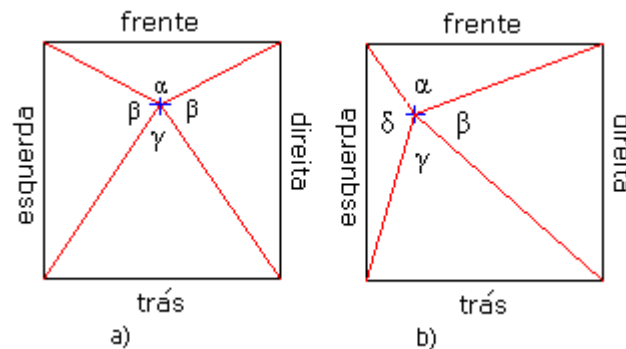


Fig. 33 – Observador andando na CAVERNA Digital

O sistema de rastreamento empregado na CAVERNA Digital é o Flock of Birds (ASCENSION TECHNOLOGY CORPORATION, 2003). Ele é um dispositivo que pode ser configurado para rastrear simultaneamente a posição e orientação de múltiplos sensores. Um transmissor envia pulsos magnéticos e cada sensor mede as características do campo magnético, calcula sua posição e orientação numa frequência de até 144 medidas por segundo, e a fornece ao computador através de uma interface de comunicação RS-232C. O transmissor, com o auxílio de um segundo transmissor, pode ser colocado até uma distância de 6,01 m dos sensores.

4.19 Trabalho Colaborativo

A arquitetura do sistema FADAS na configuração colaborativa emprega a arquitetura híbrida descrita na capítulo de fundamentação teórica desta dissertação, onde o modelo é armazenado no servidor e os comandos de controle dos usuários são enviados da estação para o servidor e retransmitidos para as estações. O acoplamento é a nível de modelo. Cada usuário tem a sua própria visão do modelo. Cada usuário pode modificar o modelo e visualizá-lo de um ponto de vista próprio. Quando um objeto é modificado os demais usuário também visualizam a alteração, desde que o objeto esteja no campo de visão deles.

Os comandos entre o servidor e estações são enviados e recebidos através do protocolo de transporte TCP/IP encapsulados na classe CWinsockUDP. A configuração de endereços IP e portas é definida em arquivos texto descrito no apêndice V desta dissertação.

O administrador, em um computador com o sistema FADAS instalado com a função de servidor, configura os usuários informando um apelido e o endereço IP de cada um. Se houver um objeto com o mesmo nome do apelido fornecido, o sistema o associa com o usuário para representá-lo. Se nenhum objeto com o mesmo nome for encontrado, o sistema adiciona automaticamente um objeto do tipo Esfera e associa-o com o usuário.

Em cada estação, o endereço IP do servidor é configurado. Cada usuário deve enviar o comando “Conectar Servidor”. Se a configuração estiver correta a conexão será estabelecida. Opcionalmente, o administrador pode desabilitar um usuário, desconectando-o e impedindo novas conexões. O diretório do sistema FADAS deve estar sincronizado em cada estação ou então o diretório localizado no servidor deve ser compartilhado para todos os usuários, sendo que neste caso, somente o administrador pode abrir, fechar e salvar os mundos virtuais.

Cada usuário tem a sua própria câmera associada ao objeto com o nome do seu apelido. O usuário visualiza os companheiros através dos objetos que os representam, incluindo um texto com o apelido localizado acima do objeto.

Os usuário podem adicionar objetos ao mundo virtual que é compartilhado com os demais usuários. Cada objeto tem o seu criador e só pode ser excluído pelo seu criador ou pelo Administrador do Sistema.

O administrador pode criar um programa e executá-lo. Neste modo de execução de programa, nenhum usuário pode adicionar objetos ou escrever comandos. Ele pode apenas navegar no mundo virtual interagindo através de dispositivos de entrada, controlando o objeto que o representa. Desta maneira, os usuários podem criar programas e executá-los em rede, executando tanto a atividade de projeto e construção de um jogo multiusuário quanto a de jogador.

Os seguintes comandos foram projetados na linguagem Fadas: Mundo:AdicionaUsuario, Mundo:RemoveUsuario, Mundo:HabilitaUsuario, Mundo:DesabilitaUsuario, Mundo:ConectaServidor.

CAPÍTULO 5

TESTES E AVALIAÇÃO DOS RESULTADOS

“Dentro de milênios eles virão nas asas do vapor, através do espaço, por sobre o oceano. Os jovens habitantes da América serão hóspedes da velha Europa. Virão aqui ver os monumentos e as cidades submersas, tal como nós, em nosso tempo, procuramos as maravilhas desaparecidas da Ásia meridional. Virão dentro de muitos mil anos.”

*“ – A Europa! É o que se ouve entre a jovem geração da América.
- As maravilhosas terras dos nossos antepassados, dos nossos sonhos e fantasias. A Europa...”**

Hans Christian Andersen

* ANDERSEN, H.C. Contos de Andersen. Trad. de Guttorm Hanssen. Rio de Janeiro: Paz e Terra, 1991.

Em sete de junho de 2003, realizaram-se testes preliminares do protótipo do sistema FADAS com quatro crianças na sala de desenvolvimento da CAVERNA Digital. As crianças tinham sete, oito, nove e doze anos, sendo que a primeira ainda não era alfabetizada e a segunda tinha acabado de ser alfabetizada. Cada criança teve acesso a um computador pessoal incluindo um *joystick*. Os testes foram conduzidos pelo autor que atuou como instrutor do sistema FADAS e foram acompanhados pelas educadoras Profa. Dra. Maria Zilda da Cunha e Profa. Dra. Roseli de Deus Lopes. A figura 34 apresenta uma fotografia com as crianças testando o sistema FADAS enquanto que a figura 35 apresenta uma fotografia com uma criança sendo entrevistada.



Fig. 34 – Primeiros usuários testando o protótipo do sistema FADAS



Fig. 35 – Pesquisadores entrevistando um usuário

5.1 Objetivo

O objetivo principal dos testes foi verificar a facilidade de uso e entendimento dos comandos básicos da linguagem Fadas. Os objetivos secundários foram verificar o interesse e a motivação das crianças, o nível de familiaridade e aceitação de interação com um mundo virtual tridimensional, as eventuais dificuldades enfrentadas e colher sugestões de melhorias e novas funcionalidades.

5.2 Roteiro

As crianças e acompanhantes chegaram no laboratório por volta de 9:30 horas. Cada criança recebeu uma tabela impressa com os comandos básicos do sistema FADAS e acomodou-se em um computador. Um dos computadores das crianças projetava a imagem no monitor de vídeo e numa tela visualizada por todos. O instrutor explicou que se tratava de um teste do sistema em desenvolvimento e não do desempenho deles (crianças), e solicitou a colaboração das crianças para testar o protótipo e fazer sugestões.

As atividades então foram efetuadas, das 10:00 às 11:00 horas, com o instrutor explicando os comandos e conceitos associados e as crianças testando os comandos em seus computadores.

A seguir, cada uma das crianças prestou o seu depoimento, depois todas crianças foram conhecer a CAVERNA Digital onde, com óculos obturadores para visão estereoscópica, passaram por cenários virtuais durante aproximadamente 20 minutos. A figura 36 apresenta cenas dos usuários interagindo na CAVERNA Digital.



Fig. 36 – Usuários interagindo com a CAVERNA Digital

5.3 Atividades

A primeira atividade foi a navegação no espaço do mundo virtual através do joystick. As crianças não tiveram dificuldades possivelmente por estarem habituadas a este tipo de dispositivo encontrado em video games.

A janela de objetos do mundo virtual foi apresentada e os comandos de movimentação e rotação foram explicados e demonstrados. As crianças conseguiram escrever seus próprios comandos, assimilando rapidamente a sintaxe dos comandos. Observou-se que as crianças freqüentemente saíam de seus computadores para verificar os comandos digitados pelos demais, curiosos por ver o que os outros conseguiam e para auxiliar o outro. Quando o instrutor perguntava o que havia de errado em um comando, as três crianças alfabetizadas, após observar atentamente, encontravam o problema.

A seguir foram ensinados os comandos de movimentação e rotação da câmera, que foram experimentados com sucesso pelas crianças. A criança não alfabetizada precisava de auxílio de um adulto para digitar os comandos.

Os próximos comandos apresentados foram esconder e mostrar objetos, diminuir e aumentar e sub-objetos. O conceito de objetos hierárquicos, com a ajuda da janela de objetos, foi facilmente assimilado. O conceito de objeto utilizado no sistema FADAS, em que um objeto pode executar ações e tem diversas características teve uma aceitação tranqüila.

O comando para alterar a característica Cor de um objeto ou sub-objeto foi apresentado e também bem assimilado. As crianças experimentaram diversos percentuais de combinação das luzes vermelha, verde e azul refletidas pelos objetos. O comando para inserir textos tridimensionais foi testado, com cada crianças escrevendo uma palavra ou frase, alterando a sua cor e efetuando rotações no espaço.

Os comandos de animação onde o usuário selecionava a animação disponível, com o auxílio da janela dedicada, gerou entusiasmos nas crianças, ansiosas por ver cada animação disponível. A combinação da animação “Correndo” com o movimento na direção “para frente” foi executado sem problemas.

O próximo passo que também gerou bastante entusiasmo e curiosidade das crianças foi o de adicionar objetos existente na biblioteca, como animais, plantas, casas e uma catedral. As crianças navegaram pelo mundo virtual visualizando os objetos incluídos, inclusive alterando a característica velocidade da câmera.

Como na viagem espacial imaginária do conto *Dentro de Milênios* de Andersen, as crianças, neste rápido teste, puderam navegar por uma catedral virtual, visualizando-a cada qual por um caminho, com uma velocidade, a sua maneira.

5.4 Depoimentos

A criança de sete anos disse “Achei legal”, enquanto que a criança de oito anos, com relação à dificuldade, disse que era “mais ou menos”.

A criança de nove anos, disse em relação ao sistema FADAS: “É bem legal”. Ela apontou um defeito: “Quando você não consegue mais ver a coisa que sumiu, falta um comando para aparecer onde que estão as coisas” “Para a câmera ir

onde está”. Com relação à dificuldade encontrada, ela disse: “Normal, não é muito difícil mas também não é muito fácil”.

A criança de doze anos sugeriu recursos que poderiam ser implementados para agilizar os comandos e que os movimentos deveriam ser mais reais: “ela vai meio que reta”. Quanto a motivação ela disse “seria mais legal se a gente pudesse montar uma mesa, uma cama”, depois “Na catedral, você vai lá e coloca um padre, uma cruz”.

A Profa. Dra. Maria Zilda da Cunha, psicopedagoga, mestra em Comunicação e Semiótica (PUC-SP) e doutora em Letras (USP), escreveu o seu depoimento sobre os testes preliminares do sistema FADAS que acompanhou:

“Acompanhar os testes preliminares trouxe a certeza de que apenas este contato com o protótipo do sistema FADAS é insuficiente para que possamos registrar aspectos que seriam verificados, se no mínimo, mais dois experimentos fossem realizados. Isto, dado o potencial educativo que se pode vislumbrar na interação da criança com esse mundo virtual.

O ambiente imersivo de informações mostrou que ler, perceber, escrever, pensar, criar e sentir podem assumir características inéditas em virtude do poder definidor do sistema - no confluir de linguagens, na hibridização de mídias que aciona, e, conseqüentemente, na mistura de sentidos que leva o receptor a interagir.

Por ser uma linguagem com hiper-sintaxes, refuncionaliza, retece e recombina linguagens em uma mesma malha multidirecional e dado o seu caráter movente submete-se às intervenções do usuário. No reticular de fluxos informacionais desenha-se uma arquitetura para a inserção interativa do leitor que ao escolher um percurso, entre outras possibilidades, estabelece sua co-participação na produção. Daí a necessidade de se realizar um mapeamento mental do desenho estrutural do documento, para obter flexibilidade no ato de processar essa leitura em trânsito.

A atenção do aprendiz, evidentemente, depende do conteúdo apresentado e do mapeamento inteligente das informações, portanto do grau de criatividade da produção hipermidiática.

Neste cenário, percebe-se o sistema Fadas não apenas como um ambiente técnico, mas vital, tendo em vista a interação e trânsito de sentidos, o acionar desafiador à inteligência e a regeneração da sensibilidade, extraindo recursos no interior das próprias redes para um renovar incessante do imaginário criador.

Pudemos observar as crianças muito à vontade com a tridimensionalidade do espaço virtual; verificamos que compreenderam sem maiores dificuldades os comandos básicos, sentiram-se desafiadas a resolverem determinados problemas propostos pelo próprio sistema e flagramos a busca de formas para construir ou reconstruir cenários. A bem da verdade, queriam povoá-los com objetos, acionando mecanismos imaginativos e de raciocínio, no embate entre o prazer e o limite da máquina, a consciência da imersão no ambiente tridimensional.

É importante salientar, a atividade com FADAS, num primeiro momento, demonstra a capacidade de as crianças compreenderem as informações sobre os comandos, de mapear a sintaxe, de interagir com curiosidade e motivação, mostrando que o sistema é desafiador. No entanto, mais que isso, verifica-se, o desenvolvimento de uma capacidade metacognitiva mais sofisticada – a de reflexão sobre a ação com a linguagem hipermidiática, na medida em que foram capazes pensar sobre as ações efetuadas e fazer algumas sugestões para alterações no projeto. Além de se mostraram capazes de avaliar e falar sobre suas performances durante o teste.

Isto revela como a produção do sistema FADAS é capaz de suscitar curiosidade, atenção, desafios cognitivos; proporcionar construção de conhecimentos.

O potencial dessa ferramenta, permite-nos projetar seu uso nas escolas, com professores que possam promover ambientes desafiadores e interativos: professor - alunos, alunos – alunos; com comandas mais

específicas, criação de jogos e trabalhos de produção e releitura de textos.”

5.5 Resultados

Os comandos básicos da linguagem Fadas foram facilmente assimilados e as crianças se sentiram à-vontade no ambiente do mundo virtual. A alfabetização se mostrou, como se esperava, um pré-requisito. A criança de oito anos, recém alfabetizada, conseguiu escrever seus comandos, inclusive corrigindo erros de sintaxe nos comandos das outras crianças.

Alguns novos comandos do sistemas FADAS foram projetados por causa das sugestões das crianças. O comando “Aponta” para apontar a câmera ou um objeto na direção de outro objeto. Os comandos “AssociaAnimaçãoComMovimento” e “AnimaçãoRepouso” foram projetados para dar maior naturalidade aos movimentos dos objetos ao executar animações quando o usuário solicita um movimento de um objeto.

A motivação e interesse foi grande. As crianças se mostraram curiosas, inclusive para ver os avanços dos demais e ajudar um ao outro a encontrar erros e executar as atividades.

5.6 Conclusão

O sistema FADAS mostrou potencial como ferramenta para crianças pela facilidade de uso e motivação verificadas no teste.

As crianças, como era de se esperar, exigiram animações nos movimentos dos objetos assim como uma biblioteca mais farta para satisfazer um pouco da sua imaginação .

No entanto, uma linguagem de programação leva tempo para ser aprendida, de modo que o sistema FADAS exige uma carga horária maior para ser assimilado. Recomenda-se que atividades dirigidas com carga horária e faixa etárias adequadas sejam feitas para que um grupo de crianças adquira o domínio dos comandos e da linguagem, assim como testes de avaliação do grau de dificuldade na interação com o ambiente imersivo da CAVERNA Digital, através de *joystick* e “varinha mágica” em conjunto com avaliações do aprendizado e motivação.

CAPÍTULO 6

CONCLUSÕES E TRABALHOS FUTUROS

*“Até que ponto e em que medida, por exemplo, um cientista, no recesso de um laboratório, engajado no avanço de aperfeiçoamento técnico desse meio, testando esse aperfeiçoamento nas aventuras de ensaio e do erro, não tem muito ou quase tudo de um artista ?”**

Lúcia Santaella

* SANTAELLA, L. *Cultura das mídias*. São Paulo: Experimento, 1996.

Neste capítulo o trabalho é concluído e são apresentadas as suas contribuições e sugestões para trabalhos futuros.

6.1 Contribuições

A principal contribuição científica inédita deste trabalho, que pretendemos ainda explorar em futuras publicações científicas, é a concepção e desenvolvimento de um ambiente de autoria para criação e manipulação de ambientes imersivos estereoscópicos de múltiplas projeções.

Acredita-se que este trabalho possa contribuir em tornar acessível os ambientes imersivos de Realidade Virtual de projeções múltiplas para usuários não especialistas como o estudante, professor ou pesquisador. Este tipo de ambiente de Realidade Virtual, que emprega diversas tecnologias, está crescendo, oferecendo cada vez mais recursos de interface para o usuário interagir com mundos virtuais e a um custo cada vez menor. O seu uso na área de educação, apesar de seu grande potencial, ainda é incipiente. Com o sistema FADAS será possível estimular a criação de novas aplicações, e enriquecer o processo de ensino/aprendizagem aliando os benefícios de um sistema de autoria com um ambiente imersivo gerado por computador, que encanta e estimula a criança.

Atualmente a criança já está habituada à tecnologia, seja por causa do video game, dos filmes de ficção, dos jogos multiusuários, dos parques temáticos e outras atividades de entretenimento. A cada momento a tecnologia está mais próxima da sociedade, em especial da criança e do adolescente. Esta tecnologia que fascina os jovens, também pode ser empregada na educação e contribuir para o processo de aprendizagem.

Os comandos básicos da linguagem de programação Fadas, mostraram-se de fácil assimilação pela criança, que também se portou naturalmente no ambiente virtual, navegando com facilidade, provavelmente por causa da influência dos jogos de video game. Foi possível criar objetos e os manipular no mundo virtual, alterando suas características e gerando movimentos animados.

O ambiente colaborativo, fornecido pelo sistema FADAS, permitirá que vários usuários compartilhem um mesmo mundo virtual, promovendo a interação entre usuários remotos, seja cooperando ou competindo uns com os outros.

Possibilitando por exemplo, que um professor, interaja remotamente com os estudantes “dentro” de um mesmo mundo virtual, podendo agir como facilitador ou agente de aprendizagem.

O sistema FADAS, concebido e implementado nesta pesquisa, foi descrito num artigo aceito para publicação e apresentação no SVR2003 – VI Symposium on Virtual Reality (GÓES; ZUFFO; LOPES, 2003).

6.2 Trabalhos Futuros

As técnicas de interação em ambiente imersivo de projeções múltiplas ainda têm um largo caminho a percorrer. O usuário, principalmente o usuário criança, exige naturalidade e poder na interface, de modo que técnicas de navegação, seleção e manipulação de objetos próximos e distantes, devem ser aperfeiçoadas. Uma técnica que parece ter um excelente potencial neste ambiente é o reconhecimento de voz.

Outras experiências devem ser realizadas para se avaliar o potencial do sistema FADAS e sua aplicabilidade na escola. Uma sugestão é explorar o livro infantil *Pretinha e Branquinha* (GÓES, 1991), que inspiraram o nome do sistema FADAS, onde os personagens fadas pintam com tinta de invisibilidade partes do corpo de outros personagens, que, por sua vez, se vingam na mesma moeda ao final da estória. Uma atividade envolvendo as crianças executando a mesma arte das personagens do livro, é possível de ser realizada dentro do ambiente imersivo da CAVERNA Digital através do sistema FADAS, utilizando-se uma varinha mágica tridimensional para selecionar e alterar as propriedades físicas das personagens do mundo virtual contado pelo livro.

Pesquisas envolvendo parcerias entre pesquisadores da área de educação, da psicologia, da literatura e da área de engenharia, podem ser feitas com o sistema FADAS no ambiente imersivo da CAVERNA Digital, analisando-se as possíveis contribuições que um sistema de autoria neste ambiente pode oferecer ao processo de aprendizagem da criança, assim como pesquisas sobre a viabilidade da introdução deste ambiente imersivo de autoria em escolas públicas e particulares do Brasil.

6.3 Conclusão

Neste trabalho propõe-se o sistema de autoria FADAS para a criação, manipulação e exploração de mundos virtuais em ambientes imersivos de Realidade Virtual. A grande vantagem do FADAS é prover uma interface simples através de dispositivos de interação e, também, uma interface de programação com comandos simples orientados a usuários não familiarizados com os aspectos tecnológicos e complexos das ferramentas de programação convencionais. Este sistema ampliará as possibilidades de acesso de usuários comuns aos sofisticados recursos da Realidade Virtual para a construção de aplicações em diversas áreas do conhecimento humano, em particular aquelas relacionadas com o aprendizado e o entretenimento.

Através do sistema FADAS, as crianças terão um estímulo a mais por tratar-se de uma proposta lúdica, interativa, que lhe permitirão sentir-se participante de um mundo virtual.

Por outro lado, os educadores terão a seu favor, o forte estímulo que na escola representará um ambiente imersivo com potencial extremamente produtivo para a aprendizagem.

Pelo aspecto inovador que soma uma ferramenta de autoria com os ambientes imersivos de Realidade Virtual, os quais também configuram uma abertura para outros trabalhos de pesquisadores, concluimos afirmando que para nós foi extremamente estimulante desenvolver esta dissertação, que contou com a clareza e discernimento dos nossos orientadores, e sobre tudo com a convicção de que a pesquisa no Brasil é uma realidade promissora e em expansão.

ANEXO A

COMANDOS DO SISTEMA FADAS

1 Ações de Objetos Tridimensionais

1.1 Move

Descrição: ação de um objeto tridimensional responsável por mover um objeto informando direção (texto), distância (número) e opcionalmente a duração em segundos (número). As seguintes direções estão disponíveis: “para frente”, “para trás”, “para esquerda”, “para direita”, “para cima”, “para baixo”, sendo que a contração “pra” pode ser utilizada. Se a duração não for informada, o movimento será imediato.

Exemplo: mover o objeto Branquinha 10 metros para a esquerda em 1 segundo.

Branquinha:Move “para esquerda”; 10; 1.

1.2 Vira

Descrição: ação de um objeto tridimensional responsável por virar um objeto informando direção (texto), ângulo (número entre -360 e +360) e opcionalmente a duração em segundos (número). As seguintes direções estão disponíveis: “para esquerda”, “para direita”.

Exemplo: virar o objeto Pretinha 90 graus para a esquerda em 5 segundos

Pretinha:Vira “para esquerda”; 90; 1.

1.3 Inclina

Descrição: ação de um objeto tridimensional responsável por inclinar um objeto informando direção (texto), ângulo (número entre -360 e +360) e opcionalmente a duração em segundos (número). As seguintes direções estão disponíveis: “para cima”, “para baixo”.

Exemplo: inclinar o objeto Pretinha 30 graus para cima em 2 segundos

Pretinha:Inclina “para cima”; 30; 2.

1.4 Angula

Descrição: ação de um objeto tridimensional responsável por angular um objeto informando direção (texto), ângulo (número entre -360 e +360) e opcionalmente a duração em segundos (número). As seguintes direções estão disponíveis: “para esquerda”, “para direita”.

Exemplo: angular o objeto Branquinha 45 graus para a esquerda em 10 segundos

Branquinha:Angula “para esquerda”; 45; 10.

1.5 Aumenta

Descrição: ação de um objeto tridimensional responsável por aumentar as dimensões de um objeto informando o percentual (número) a ser aumentado, entre 1 e 1000, e opcionalmente a duração em segundos (número).

Exemplo: aumentar o tamanho do objeto a Pretinha em 10% em 5 segundos.

Pretinha:Aumenta 10;5.

1.6 Diminui

Descrição: ação de um objeto tridimensional responsável por diminuir as dimensões um objeto informando o percentual (número) a ser diminuído, entre 1 e 99, e opcionalmente a duração em segundos (número).

Exemplo: diminuir o objeto Pretinha em 50% em 1 segundo.

Pretinha:Diminui 50;1.

1.7 Aparece

Descrição: ação de um objeto tridimensional responsável por mostrar um objeto escondido.

Exemplo: mostrar o sub-objeto “BraçoDireito” do objeto Branquinha.

Branquinha: BraçoDireito: Aparece.

1.8 Esconde

Descrição: ação de um objeto tridimensional responsável por esconder um objeto.

Exemplo: esconder o objeto Branquinha.

Branquinha: Esconde.

1.9 Achata

Descrição: ação de um objeto tridimensional responsável por achatar um objeto em uma determinada direção, informando a direção, o percentual entre 1 e 99 e opcionalmente a duração em segundos. As direções disponíveis são as seguintes: “para cima e para baixo”, “para frente e para trás” e “para esquerda e para direita”.

Exemplo: achata o objeto Pretinha em 50%, na direção “para cima e para baixo”, durante 5 segundos.

Pretinha: Achata “para cima e para baixo”, 50;5.

1.10 Estica

Descrição: ação de um objeto tridimensional responsável por esticar um objeto em uma determinada direção, informando a direção, o percentual entre 1 e 1000 e opcionalmente a duração em segundos. As direções disponíveis são as seguintes: “para cima e para baixo”, “para frente e para trás” e “para esquerda e para direita”.

Exemplo: estica o objeto Pretinha em 100%, na direção “para cima e para baixo”, durante 5 segundos

Pretinha: Estica “para cima e para baixo”, 100;5.

1.11 Entorta

Descrição: ação de um objeto tridimensional responsável por entortar um objeto em uma determinada direção linearmente proporcional a altura do objeto, informando o percentual de proporção com a altura e opcionalmente a duração em segundos. As direções disponíveis são as seguintes: “para frente”, “para trás”, “para direita” e “para esquerda”.

Exemplo: entorta o objeto Branquinha na direção “para frente”, 100% proporcional a altura, durante 5 segundos.

Branquinha:Entorta “para frente”;100;5.

2 Características de Objetos Tridimensionais

2.1 Nome

Descrição: característica de um objeto tridimensional utilizada para definir um novo nome para o objeto.

Exemplo: trocar o nome do objeto de nome Objeto3 para Urso.

Objeto3:Nome = “Urso”.

2.2 Nome Pai

Descrição: característica de um objeto tridimensional utilizada para definir o nome do objeto Pai de um objeto.

Exemplo: definir que o pai do objeto Roda é o objeto Carro.

Roda:NomePai = “Carro”.

2.3 Cor

Descrição: característica de um objeto tridimensional utilizada para definir a cor de um objeto tridimensional ou a cor de fundo do objeto Mundo, ou seja, as componentes de luz vermelho, verde e azul que o objeto reflete. Esta característica

define uma mesma cor para as luzes ambiente, difusa e especular refletidas pelo objeto.

Exemplo: a cor do sub-objeto Rosto do objeto Branquinha é a combinação de 80% de luz vermelha, 30% de luz verde e 45% de luz azul.

Branquinha:Rosto:Cor = C(80; 30;45).

2.4 CorLuzAmbiente

Descrição: característica de um objeto tridimensional utilizada para definir a luz ambiente refletida pelo objeto.

Exemplo: define a cor da luz ambiente refletida pelo objeto Cubo como 80% de vermelho.

Cubo:CorLuzAmbiente = C(80;0;0).

2.5 CorLuzDifusa

Descrição: característica de um objeto tridimensional utilizada para definir a luz difusa refletida pelo objeto.

Exemplo: define a cor da luz difusa refletida pelo objeto Cubo como 80% de vermelho, 10% de verde e 30% de azul.

Cubo:CorLuzDifusa = C(80;10;30).

2.6 CorLuzEspecular

Descrição: característica de um objeto tridimensional utilizada para definir a luz especular refletida pelo objeto.

Exemplo: define a cor da luz especular refletida pelo objeto Esfera como 10% de vermelho, 20% de verde e 30% de azul.

Esfera:CorLuzEspecular = C(10;20;30).

2.7 LuzEmissiva

Descrição: característica de um objeto tridimensional utilizada para simular a luz emitida por um objeto. Esta característica não afeta os demais objetos. Para simular uma luz posicional, é preciso criar uma fonte de luz.

Exemplo: define a cor da luz “emitida” pelo objeto Lâmpada como 70% de vermelho.

Lâmpada:LuzEmissiva = C(70;0;0).

2.8 Opacidade

Descrição: característica de um objeto tridimensional utilizada para definir o percentual de opacidade de um objeto.

Exemplo: o objeto carro é 50% opaco

Carro:Opacidade = 50.

2.9 Velocidade

Descrição: característica de um objeto tridimensional utilizada para definir a velocidade de um objeto no direção “para frente”.

Exemplo: define a velocidade do objeto Urso como 5 unidades por segundo na direção “para frente”.

Urso:Velocidade = 5.

2.10 RotacaoAfetaDirecaoVelocidade

Descrição: característica de um objeto tridimensional utilizada para definir se uma rotação afeta a direção atual da velocidade de um objeto. O número 1 é o valor padrão e indica que a direção da velocidade é alterada quando uma rotação é executada no objeto. O valor 0 indica que as rotações não afetam a direção da velocidade atual do objeto.

Exemplo: define que a direção da velocidade do objeto Bola não deve ser afetada pelas rotações.

Bola:RotacaoAfetaDirecaoVelocidade = 0.

2.11 Aceleração

Descrição: característica de um objeto tridimensional utilizada para definir a aceleração de um objeto na direção “para frente”.

Exemplo: define a aceleração do objeto Pretinha como 2 unidades por segundo quadrado.

Pretinha:Aceleração = 2.

2.12 Coeficiente de Atrito

Descrição: característica de um objeto tridimensional utilizada para simular o coeficiente de atrito de um objeto. O coeficiente pode ser um valor entre zero e cem. O valor padrão é 0 de modo que quando uma velocidade é atribuída a um objeto, ela permanece constante permanentemente. Quando um coeficiente de atrito é definido para um objeto, uma aceleração no sentido contrário a velocidade do objeto age sobre o objeto reduzindo gradativamente a velocidade até atingir zero.

Exemplo: o coeficiente de atrito do objeto Pretinha é 50.

Pretinha:CoeficienteAtrito = 50.

3 Animações

3.1 Anima

Descrição: ação de um objeto tridimensional responsável por executar uma animação informando o tempo total da animação e opcionalmente a duração de um ciclo. Se a duração do ciclo não for informada, o sistema utilizará a última duração de ciclo informada ou 1 segundo no caso de uma duração não ter sido informada anteriormente.

Exemplo: Execução da animação de nome “Andando” do objeto Ogro durante 10 segundos, com um ciclo de 2 segundos.

Ogro:Anima “Andando”;10;2.

3.2 AssociaAnimacaoComMovimento

Descrição: ação de um objeto tridimensional responsável por associar uma animação de um objeto com um movimento em determinada direção, de modo que quando o objeto se movimentar, a animação será executada. A duração do ciclo será a última duração de ciclo informada ou 1 segundo no caso desta duração não ter sido informada anteriormente.

Exemplo: associa a animação “Andando” do objeto Ogro ao movimento na direção “Para frente”. O segundo comando solicita que o objeto se mova para frente, de modo que a animação “Andando” será executada automaticamente durante este movimento.

Ogro:AssociaAnimacaoComMovimento “Andando”;“Para frente”.

Ogro:Move “para frente”;10;5.

3.3 AnimaçãoRepouso

Descrição: característica de um objeto tridimensional utilizada para definir a animação de repouso para um objeto a ser executada quando este estiver parado por um tempo determinado

Exemplo: define que a animação “Pensando” do objeto Ogro será executada sempre que o objeto estiver parado por um tempo maior que o tempo de espera.

Ogro:AnimaçãoRepouso = “Pensando”.

3.4 TempoRepouso

Descrição: característica de um objeto tridimensional utilizada para definir o tempo de repouso em minutos para um objeto. Após este tempo, o sistema executará a animação de repouso. O valor do tempo deve estar entre 1 e 30. O valor 0 desativa o controle de tempo de repouso para execução de animação.

Exemplo: define o tempo de repouso de 1 minuto para o objeto Ogro. A animação de repouso do objeto Ogro será executada sempre que o objeto estiver parado por mais de 1 minuto.

Ogro:TempoRepouso = 1.

3.5 Espera

Descrição: ação do objeto Mundo solicitando ao sistema Fadas que espere um determinado tempo (número) em segundos para executar o próximo comando. Opcionalmente o nome (texto) de uma rotina pode ser informado e neste caso o sistema continua executando os próximos comandos e quando o tempo determinado for atingido, a rotina informada é executada.

Exemplo: o sistema executa o movimento do objeto Branquinha e do objeto Pretinha e espera 10 segundos antes de executar o próximo comando que vira o objeto Branquinha para a esquerda.

Branquinha:Move "para frente";20;10.

Pretinha:Move "para frente";20;10.

Mundo:Espera 10.

Branquinha:Vira "para esquerda";90;3.

Exemplo: o sistema executa o movimento do objeto Pretinha e inicia uma contagem de 10 segundos. A seguir o sistema executa o próximo comando que move o objeto Bola. Quando a contagem chegar em 10 segundos, o sistema executa a rotina de nome “ContinuaPretinha”.

Pretinha:Move "para frente";20;10.

Mundo:Espera 10; “ContinuaPretinha”.

Bola:Move "para frente";5;10.

4 Criação de objetos sólidos

O sistema Fadas permite ao usuário a criação de texto tridimensionais e dos sólidos esfera, paralelepípedo, cubo, cilindro, cone e anel. A seguir descreve-se as ações do objeto Mundo responsáveis por criar estes objetos.

4.1 CaixaTexto3D

4.1.1 CriaCaixaTexto3D

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo CaixaTexto3D informando o nome do texto, o nome do objeto onde o texto será criado, o texto e opcionalmente o nome do fonte do texto.

Exemplo: cria um objeto do tipo CaixaTexto3D com o nome txtOla, na posição do objeto Pretinha, com o texto “Ola” e o fonte Arial. O segundo comando altera a cor do texto e o terceiro vira o texto 360 graus para a esquerda durante 10 segundos.

Mundo:CriaCaixaTexto3D “txtOla”;“Pretinha”;“Olá”; “Arial”.

txtOla:Cor = C(100;50;50).

txtOla:Vira “para esquerda”;360;10.

4.1.2 Texto

Descrição: característica do objeto do tipo CaixaTexto3D que define o texto.

Exemplo: altera o texto da caixa de texto txtOla para “Olá Júnior”.

txtOla:Texto = “Olá Júnior”.

4.2 CriaEsfera

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Esfera, informando o nome do objeto, o nome do objeto onde a esfera será criada e o raio.

Exemplo: cria um objeto do tipo Esfera com o nome Sol, na posição do objeto Pretinha, com raio de 10 unidades.

Mundo:CriaEsfera “Sol”;“Pretinha”;10.

4.3 CriaParalelepipedo

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Paralelepípedo, informando o nome do objeto, o nome do objeto onde o paralelepípedo será criado, a largura, altura e profundidade.

Exemplo: cria um objeto do tipo Paralelepípedo na posição do objeto Branquinha, com nome Pedra, largura 10, altura 20 e profundidade de 30 unidades.

Mundo:CriaParalelepipedo “Pedra”;“Branquinha”10;20;30.

4.4 CriaCubo

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Cubo, informando o nome do objeto, o nome do objeto onde o cubo será criado e a altura do cubo.

Exemplo: cria na posição do objeto Pretinha, um objeto do tipo Cubo com nome Dado e altura de 10 unidades.

Mundo:CriaEsfera “Dado”;“Pretinha:”;10.

4.5 CriaCilindro

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Cilindro, informando o nome do objeto, o nome do objeto onde o cilindro será criado, a altura e o raio do cilindro.

Exemplo: cria na posição do objeto Pretinha, um cilindro com nome Cilindro1, altura 20 e raio de 10 unidades.

Mundo:CriaCilindro “Cilindro1”;“Pretinha”;20;10.

4.6 CriaCone

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Cone, informando o nome do objeto, o nome do objeto onde o cone será criado, a altura e raio da base do cone.

Exemplo: cria na posição do objeto Pretinha, um cone com nome Cone, altura 20 e raio da base 10.

Mundo:CriaCone “Cone”;“Pretinha”;20;10.

4.7 CriaAnel

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Anel, informando o nome do objeto, o nome do objeto onde o anel será criado, o raio interno e externo. A diferença entre o raio externo e o raio interno equivale ao diâmetro da espessura do anel.

Exemplo: cria um objeto do tipo anel com nome Anel, raio interno 10 e raio externo 20, na posição do objeto Pretinha.

Mundo:CriaAnel “Anel”;“Pretinha”;10;20.

5 Criação de pontos, linhas e polígonos

O sistema Fadas permite ao usuário a criação de pontos através de objetos do tipo Ponto, linhas (pontos interligados) através de objetos do tipo PontosLigados e polígonos (polígonos convexos formados por um conjunto de triângulos, quadriláteros ou vértices) através de objetos do tipo Polígono. As ações do objeto Mundo responsáveis por criar estes objetos são descritas a seguir.

5.1 CriaPonto

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo Ponto, informando o nome (texto) do objeto e o nome (texto) do objeto onde o ponto será criado.

Exemplo: cria na posição do objeto Branquinha, um ponto com o nome Ponto1.

Mundo:CriaPonto “Ponto1”;“Branquinha”.

5.2 PontosLigados

Um objeto do tipo PontosLigados é um polígono formado por um conjunto de pontos interligados. A cada ponto adicionado, o sistema liga o novo vértice ao último vértice do conjunto.

5.2.1 CriaPontosLigados

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo PontosLigados informando o nome (texto) do objeto. Após criar o objeto, o usuário deve adicionar novos pontos através da ação AdicionaPonto.

Exemplo: cria um objeto do tipo PontosLigados com o nome ptosSeta.

Mundo:CriaPontosLigados “ptosSeta”.

5.2.2 AdicionaPonto

Descrição: ação de um objeto do tipo PontosLigados responsável por adicionar um ponto a um objeto do tipo PontosLigados, informando o nome (texto) do objeto onde o ponto será criado.

Exemplo: adiciona novos pontos ao objeto ptosSeta de acordo com a posição do objeto Pretinha.

ptosSeta:AdicionaPonto “Pretinha”.

Pretinha:Move “para esquerda”;5.

ptosSeta:AdicionaPonto “Pretinha”.

Pretinha:Move “para cima”;5.

ptosSeta:AdicionaPonto “Pretinha”.

5.2.3 RemoveUltimoPonto

Descrição: ação de um objeto do tipo PontosLigados responsável por remover o último ponto de um objeto do tipo PontosLigados.

Exemplo: remove o último ponto do objeto de nome ptosSeta.

ptosSeta:RemoveUltimoPonto.

5.3 Polígono

5.3.1 CriaPoligonoTriangulos

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo PoligonoTriangulos. Este objeto é um polígono convexo formado por um conjunto de faces triangulares. O nome (texto) do objeto deve ser informado, sendo que os vértices devem ser adicionados na ordem anti-horária através da ação AdicionaVertice.

Exemplo: cria um objeto do tipo PoligonoTriangulos com o nome Pirâmide.

Mundo:CriaPoligonoTriangulos “Pirâmide”.

5.3.2 CriaPoligonoQuadrilatero

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo PoligonoQuadrilatero. Este objeto é um polígono convexo formado por um conjunto de faces quadriláteras. O nome (texto) do objeto deve ser informado, sendo que os vértices devem ser adicionados na ordem anti-horária através da ação AdicionaVertice.

Exemplo: cria um objeto do tipo PoligonoQuadrilatero com o nome Cubo.

Mundo:CriaPoligonoQuadrilatero “Cubo”.

5.3.3 CriaPoligonoVertices

Descrição: ação do objeto Mundo responsável por criar um objeto do tipo PoligonoVertices. Este objeto é um polígono convexo formado por um conjunto de vértices. O nome (texto) do objeto deve ser informado, sendo que os vértices devem ser adicionados na ordem anti-horária através da ação AdicionaVertice. O polígono deve ter 3 ou mais vértices sendo que o último vértice é ligado ao primeiro.

Exemplo: cria um objeto do tipo PoligonoVertices com o nome Pirâmide.

Mundo:CriaPoligonoVertices “Pirâmide”.

5.3.4 AdicionaVertice

Descrição: ação de objeto dos tipos PoligonoTriangulos, PoligonoQuadrados ou Polígono, responsável por adicionar um vértice a face do polígono, informando o nome (texto) do objeto onde o vértice será criado. Os vértices devem ser adicionados na ordem anti-horária

Exemplo: adiciona novos vértices ao objeto Pirâmide de acordo com a posição do objeto Pretinha.

Pirâmide:AdicionaVertice “Pretinha”.

Pretinha:Move “para esquerda”;10.

Pirâmide:AdicionaVertice “Pretinha”.

Pretinha:Move “para cima”;10.

Pretinha:Move “para direita”;5.

Pirâmide:AdicionaVertice “Pretinha”.

5.3.5 RemoveUltimoVertice

Descrição: ação de objeto dos tipos PoligonoTriangulos, PoligonoQuadrados ou Polígono, responsável por remover o último vértice adicionado.

Exemplo: remove o último vértice do objeto de nome Pirâmide.

Pirâmide:RemoveUltimoVertice.

6 Adição e seleção de objetos tridimensionais

A seguir descreve-se as ações do objeto Mundo responsáveis por adicionar e remover e selecionar objetos tridimensionais no mundo virtual.

6.1 AdicionaObjeto

Descrição: ação do objeto Mundo responsável por adicionar um objeto do ao mundo virtual, informando o caminho (texto) do arquivo de objeto fadas onde se localiza o modelo tridimensional e opcionalmente o nome (texto) do objeto e o nome do objeto sobre o qual o novo objeto será posicionado e orientado.

Se um nome para o objeto não for informado, o sistema utilizará o nome gravado arquivo do modelo ou, na ausência deste, atribuirá um nome resultante da concatenação da palavra “objeto” e de um número seqüencia.

Se o nome do objeto referência sobre o qual o novo objeto será posicionado e orientado não for informado, o sistema posicionará o novo objeto na posição central do mundo virtual com os eixos locais alinhados aos eixos globais.

Exemplo: adiciona o modelo 3D como o nome Perola a partir do arquivo Fusca.FOB
Mundo:AdicionaObjeto “c:\fadas\biblioteca\veiculo\Fusca.FOB”; “Perola”.

Exemplo: adiciona o modelo 3D como o nome Carro a partir do arquivo CarroCorrida.FOB na posição e orientação do objeto Pretinha.

Mundo:AdicionaObjeto “c:\fadas\biblioteca\veiculo\CarroCorrida.FOB”;
“Carro”;“Pretinha”.

6.2 RemoveObjeto

Descrição: ação do objeto Mundo responsável por remover um objeto do mundo virtual, informando o nome (texto) do objeto.

Exemplo: remove o objeto Perola

Mundo:RemoveObjeto “Perola”.

6.3 ImportaObjeto

Descrição: ação do objeto Mundo responsável por adicionar um objeto do ao mundo virtual, informando o caminho (texto) do arquivo de objeto ASE ou MD2 onde se localiza o modelo tridimensional, e opcionalmente o nome (texto) do objeto.

Exemplo: adiciona o modelo 3D a partir do arquivo avião.ase, com o nome Avião

Mundo:ImportaObjeto “c:\fadas\biblioteca\veiculo\aviao.ase”;“Avião”.

6.4 SeleccionaObjeto

Descrição: ação do objeto mundo responsável por selecionar um objeto, para permitir a execução de comandos sobre o objeto selecionado. É necessário informar o nome (texto) do objeto a ser selecionado.

Exemplo: seleciona o objeto Perola
Mundo:SeleccionaObjeto “Pretinha”.

7 Câmeras

7.1 AdicionaCamera

Descrição: ação do objeto Mundo responsável por adicionar uma câmera ao mundo virtual, informando o nome da nova câmera e o nome do objeto onde a nova câmera será posicionada e orientada

Exemplo: cria uma câmera com o nome SegundaCamera na posição e orientação do objeto Pretinha.

Mundo:AdicionaCâmera “SegundaCamera”, “Pretinha”.

7.2 AtivaCamera

Descrição: característica do objeto Mundo responsável por definir a câmera ativa. Esta característica contém o nome (texto) da câmera ativa.

Exemplo: define a câmera de nome SegundaCamera como a ativa do mundo virtual.

Mundo:CameraAtiva = “SegundaCamera”.

7.3 TrocaCamera

Descrição: ação do objeto Mundo responsável por trocar a câmera ativa pela próxima câmera da lista de câmeras do mundo.

Exemplo: troca a câmera ativa pela próxima câmera.

Mundo:TrocaCamera.

7.4 Associa

Descrição: característica de um objeto do tipo Câmera responsável por associar uma câmera a um objeto. Esta característica contém o nome (texto) da objeto associado a câmera.

Exemplo: Associa a câmera SegundaCamera com o objeto Pretinha.

SegundaCamera:Associa = “Pretinha”.

7.5 RemoveCamera

Descrição: ação do objeto Mundo responsável por remover uma câmera do mundo, informando o nome da câmera a ser removida.

Exemplo: remove a câmera SegundaCamera do mundo virtual.

Mundo:RemoveCamera “SegundaCamera”.

7.6 AumentaDeslocamentoPadrao

Descrição: ação de um objeto do tipo Câmera responsável por incrementar o deslocamento padrão de uma câmera. O deslocamento padrão é utilizado para a navegação através de joystick, teclado e mouse. Quanto maior ele for, maior será a distância deslocada em cada movimento da câmera.

Exemplo: aumenta o deslocamento padrão da câmera de nome Câmera

Câmera:AumentaDeslocamentoPadrao.

7.7 DiminuiDeslocamentoPadrao

Descrição: ação de um objeto do tipo Câmera responsável por decrementar o deslocamento padrão de uma câmera. O deslocamento padrão é utilizado para a navegação através de joystick, teclado e mouse. Quanto menor ele for, menor será a distância deslocada em cada movimento da câmera.

Exemplo: diminui o deslocamento padrão da câmera de nome Câmera

Câmera:DiminuiDeslocamentoPadrao.

7.8 AumentaRotacaoPadrao

Descrição: ação de um objeto do tipo Câmera responsável por incrementar o ângulo de rotação padrão de uma câmera. Este ângulo é utilizado para a navegação através de joystick, teclado e mouse.

Exemplo: aumenta o ângulo de rotação padrão da câmera de nome Câmera

Câmera:AumentaRotacaoPadrao.

7.9 DiminuiRotacaoPadrao

Descrição: ação de um objeto do tipo Câmera responsável por decrementar o ângulo de rotação padrão de uma câmera. Este ângulo é utilizado para a navegação através de joystick, teclado e mouse.

Exemplo: diminui o ângulo de rotação padrão da câmera de nome Câmera

Câmera:DiminuiRotacaoPadrao.

8 Fontes de Luz

8.1 AdicionaFonteLuzPosicional

Descrição: ação do objeto Mundo responsável por adiciona uma fonte de luz posicional, informando o nome (texto) da nova fonte de luz, o nome (texto) do objeto onde a fonte de luz será posicionada e as cores das componentes luz ambiente, difusa e especular.

Exemplo: cria uma fonte de luz de nome Lâmpada na posição do objeto Pretinha, sem emitir componente de luz ambiente, e emitindo luz branca para as componentes de luz difusa e especular.

Mundo:AdicionaFonteLuzPosicional "Lâmpada"; "Pretinha"; C(0;0;0); C(100;100;100); C(100;100;100).

8.2 AdicionaFonteLuzDirecional

Descrição: ação do objeto Mundo responsável por adicionar uma fonte de luz direcional, informando o nome da nova fonte de luz, o nome do objeto com a orientação para a nova fonte de luz e as componentes luz ambiente, difusa e especular.

Exemplo: cria uma fonte de luz de nome Lâmpada na direção “para frente” do objeto Pretinha, sem emitir componente de luz ambiente, e emitindo luz branca para as componentes de luz difusa e especular

Mundo:AdicionaFonteLuzPosicional “Lâmpada”; “Pretinha”; C(0;0;0); C(100;100;100); C(100;100;100).

8.3 LuzAmbiente

Descrição: característica de uma fonte de luz responsável pela componente de luz ambiente.

Exemplo: altera a luz ambiente da fonte de luz Lâmpada.

Lâmpada:LuzAmbiente = C(100;100;100).

8.4 LuzDifusa

Descrição: característica de uma fonte de luz responsável pela componente de luz ambiente.

Exemplo: altera a luz difusa da fonte de luz Lâmpada.

Lâmpada:LuzDifusa = C(100;100;100).

8.5 LuzEspecular

Descrição: característica de uma fonte de luz responsável pela componente de luz ambiente.

Exemplo: altera a luz especular da fonte de luz Lâmpada.

Lâmpada:LuzEspecular = C(100;100;100).

8.6 Liga

Descrição: ação de uma fonte de luz responsável por habilitar uma fonte de luz

Exemplo: habilita a fonte de luz Lâmpada

Lâmpada:Liga.

8.7 Desliga

Descrição: ação de uma fonte de luz responsável por desabilitar uma fonte de luz

Exemplo: desabilita a fonte de luz Lâmpada

Lâmpada:Desliga.

8.8 RemoveFonteLuz

Descrição: ação do objeto Mundo responsável por remover uma fonte de luz.

Exemplo: remove a fonte de luz Lâmpada

Mundo:RemoveFonteLuz “Lâmpada”

9 Som

9.1 AdicionaSom

Descrição: ação do objeto Mundo responsável por adicionar um arquivo de som (.WAV) ou de música (.MID), informando o caminho (texto) do arquivo e o nome (texto) do novo som.

Exemplo: adiciona o arquivo de som tenis.wav e definindo o nome somTenis

Mundo:AdicionaSom “C:\Fadas\Biblioteca\Som\tenis.wav”; “somTenis”.

9.2 Toca

Descrição: ação de um som responsável por reproduzir o som.

Exemplo: toca o som somTennis

somTennis:Toca.

9.3 Para

Descrição: ação de um som responsável por para a reprodução de um som.

Exemplo: para de tocar o som somSabia

somSabia:Para.

9.4 RemoveSom

Descrição: ação do objeto Mundo responsável por remover um som, informando o nome do som.

Exemplo: remove o som somTennis

Mundo:RemoveSom “somTennis”.

9.5 AdicionaSom3D

Descrição: ação do objeto Mundo responsável por adicionar um arquivo de som (.WAV) ou de música (.MID), informando o caminho (texto) do arquivo, o nome (texto) do novo som e o nome de um objeto com a posição e orientação do novo som. Um som 3D tem posição e pode ter orientação e velocidade.

Exemplo: adiciona um som 3D na posição e orientação do objeto Branquinha, com o nome somTennis, carregando o arquivo tenis.wav,

Mundo:AdicionaSom3D “C:\Fadas\Biblioteca\Som\tenis.wav”; “somTennis”;
“Branquinha”.

9.6 Velocidade

Descrição: característica de um som3D que define a velocidade (número) do som.

Exemplo: define a velocidade do som3D som Tennis com 5 unidades por segundo.

somTennis:Velocidade = 5.

9.7 DistanciaMinima

Descrição: característica de um som3D que define a distância mínima entre o som3D e a câmera ativa. A distância mínima, cujo valor padrão é 1, é aquela em que o volume do som está com a intensidade máxima. Quando a câmera ativa está distante o dobro da distância mínima, o volume do som é reduzido a metade, e assim por diante.

Exemplo: define a distância mínima do som3D som Tennis como 5 unidades.
somTennis:DistanciaMinima = 5.

9.8 DistanciaMaxima

Descrição: característica de um som3D que define a distância máxima entre o som3D e a câmera ativa. A distância máxima define a distância em que o som não mais será reproduzido.

Exemplo: define a distância máxima do som3D som Tennis como 50 unidades.
somTennis:DistanciaMaxima = 50.

10 Manipulação de Mundos

10.1 Salva

Descrição: ação do objeto Mundo responsável por salvar o mundo atual, informando o nome (texto) do arquivo a ser salvo.

Exemplo: salva o mundo virtual no arquivo teste.fap
Mundo:Salva "c:\fadas\mundos\Mundo2\Teste.FAP".

10.2 Abre

Descrição: ação do objeto Mundo responsável por abrir ou carregar um mundo Fadas previamente salvo, informando o caminho (texto) do arquivo a abrir.

Exemplo: abre o mundo salvo no arquivo mundo1.fap

Mundo:Abre “c:\fadas\mundos\Mundo1\Mundo1.FAP”.

10.3 Fecha

Descrição: ação do objeto Mundo responsável por fechar o mundo virtual atual.

Exemplo: fecha o mundo atual

Mundo:Fecha.

10.4 Cria

Descrição: ação do objeto Mundo responsável por criar um novo mundo virtual. Esta ação fecha o mundo atual permitindo que este seja salvo.

Exemplo: fecha o mundo atual e cria um novo mundo

Mundo:Cria.

11 Colaboração

11.1 AdicionaUsuario

Descrição: ação do objeto Mundo responsável por adicionar um novo usuário para compartilhar e colaborar no mundo virtual atual. É necessário informar o endereço IP (texto) do computador do usuário e o apelido (texto) do usuário.

Exemplo: adiciona o usuário do IP 192.168.0.40 de apelido Junior.

Mundo:AdicionaUsuario “192.168.0.40”; “Junior”.

11.2 RemoveUsuario

Descrição: ação do objeto Mundo responsável por remover um usuário do mundo virtual atual.

Exemplo: remove o usuário Junior.

Mundo:RemoveUsuario “Junior”.

11.3 HabilitaUsuario

Descrição: ação do objeto Mundo responsável por habilitar um usuário do mundo virtual atual, informando o apelido (texto) do usuário.

Exemplo: habilita o usuário Junior

Mundo:HabilitaUsuario “Junior”.

11.4 DesabilitaUsuario

Descrição: ação do objeto Mundo responsável por desabilitar um usuário do mundo virtual atual, informando o apelido (texto) do usuário.

Exemplo: desabilita o usuário Junior

Mundo:DesabilitaUsuario “Junior”.

11.5 ConectaServidor

Descrição: ação do objeto Mundo responsável por conectar um usuário ao mundo virtual atual.

Exemplo: usuário solicita conexão com o servidor Fadas.

Mundo:ConectaServidor

12 Configuração

12.1 Configuração Estereoscópica

12.1.1 AtivaEstereo

Descrição: ação do objeto Config responsável por habilitar a geração de imagens estereoscópicas.

Exemplo: habilita geração de imagens estereoscópicas.

Config:AtivaEstereo.

12.1.2 DestivaEstereo

Descrição: ação do objeto Config responsável por deabilitar a geração de imagens estereoscópicas.

Exemplo: desabilita geração de imagens estereoscópicas.

Config:DesativaEstereo.

12.1.3 AumentaDistanciaCameras

Descrição: ação do objeto Config responsável por incrementar a distância entre as câmeras utilizadas na geração de imagens estereoscópicas.

Exemplo: aumenta distância entre as câmeras.

Config:AumentaDistanciaCameras.

12.1.4 DiminuiDistanciaCameras

Descrição: ação do objeto Config responsável por decrementar a distância entre as câmeras utilizadas na geração de imagens estereoscópicas.

Exemplo: diminui distância entre as câmeras.

Config:DiminuiDistanciaCameras.

12.1.5 AumentaHIT

Descrição: ação do objeto Config responsável por incrementar a translação horizontal da imagem utilizada na geração de imagens estereoscópicas

Exemplo: aumenta a distância HIT.

Config:AumentaHIT.

12.1.6 DiminuiHIT

Descrição: ação do objeto Config responsável por decrementar a translação horizontal da imagem utilizada na geração de imagens estereoscópicas

Exemplo: diminui a distância HIT.

Config:DiminuiHIT.

12.2 Configuração de Rastreamento do Observador

12.2.1 AtivaRastreamentoObservador

Descrição: ação do objeto Config responsável por iniciar o rastreamento da posição e orientação do observador em um ambiente imersivo de projeções múltiplas. O sistema associará a posição e orientação da câmera principal do mundo virtual com a posição central do ambiente imersivo de projeções múltiplas.

Exemplo: inicia rastreamento do observador.

Config:AtivaRastreamentoObservador.

12.2.2 DesativaRastreamentoObservador

Descrição: ação do objeto Config responsável por desativar o rastreamento da posição e orientação do observador em um ambiente imersivo de projeções múltiplas.

Exemplo: desativa o rastreamento do observador.

Config:DestivaRastreamentoObservador.

12.3 Configuração de Rastreamento da Vara de Seleção

12.3.1 CriaVaraSelecao

Descrição: ação do objeto Mundo responsável por iniciar o rastreamento da posição e orientação da vara de seleção ou “varinha mágica” do usuário dentro de um ambiente imersivo de projeções múltiplas. O sistema criará um objeto virtual do tipo VaraSelecao na posição e orientação do objeto informado.

Exemplo: cria vara de seleção na posição e orientação do objeto Branquinha.

Mundo:CriaVaraSelecao “Branquinha”.

12.3.2 AumentaVaraSelecao

Descrição: ação do objeto Mundo responsável por aumentar o tamanho da vara de seleção ou “varinha mágica”, informando o nome do objeto a quem a vara pertence.

Exemplo: aumenta a vara de seleção do objeto Branquinha.

Mundo:AumentaVaraSelecao “Branquinha”.

12.2.3 DiminuiVaraSelecao

Descrição: ação do objeto Mundo responsável por diminuir o tamanho da vara de seleção ou “varinha mágica”, informando o nome do objeto a quem a vara pertence.

Exemplo: diminui a vara de seleção do objeto Branquinha.

Mundo:DiminuiVaraSelecao “Branquinha”.

12.3.4 RemoveVaraSelecao

Descrição: ação do objeto Mundo responsável por remover a vara de seleção ou “varinha mágica”, informando o nome do objeto a quem a vara pertence.

Exemplo: aumenta a vara de seleção do objeto Branquinha.

Mundo:RemoveVaraSelecao “Branquinha”.

13 Síntese dos comandos do sistema FADAS

A tabela IV sintetiza os comandos do sistema FADAS.

Tabela IV – Síntese dos comandos do sistema FADAS

Ações de Objetos Tridimensionais	Move Vira Inclina Angula Aumenta Diminui Aparece Esconde Achata Estica Entorta
Características de Objetos Tridimensionais	Nome Nome Pai Cor CorLuzAmbiente CorLuzDifusa CorLuzEspecular LuzEmissiva Opacidade Velocidade RotacaoAfetaDirecaoVelocidade Aceleração Coeficiente de Atrito
Animações	Anima AssociaAnimacaoComMovimento AnimaçãoRepouso TempoRepouso Espera
Criação de Objetos Sólidos	CaixaTexto3D - CriaCaixaTexto3D - Texto CriaEsfera CriaParalelepipedo CriaCubo CriaCilindro CriaCone CriaAnel
Criação de Pontos, Linhas e Polígonos	CriaPonto PontosLigados - CriaPontosLigados - AdicionaPonto - RemoveUltimoPonto Polígono - CriaPoligonoTriangulos - CriaPoligonoQuadrilatero - CriaPoligonoVertices - AdicionaVertice - RemoveUltimoVertice
Adição e Seleção de Objetos Tridimensionais	AdicionaObjeto RemoveObjeto ImportaObjeto SelecionaObjeto

Câmeras	AdicionaCamera AtivaCamera TrocaCamera Associa RemoveCamera AumentaDeslocamentoPadrao DiminuiDeslocamentoPadrao AumentaRotacaoPadrao DiminuiRotacaoPadrao
Fontes de Luz	AdicionaFonteLuzPosicional AdicionaFonteLuzDirecional LuzAmbiente LuzDifusa LuzEspecular Liga Desliga RemoveFonteLuz
Som	AdicionaSom Toca Para RemoveSom AdicionaSom3D Velocidade DistanciaMinima DistanciaMaxima
Manipulação de Mundos	Salva Abre Fecha Cria
Colaboração	AdicionaUsuario RemoveUsuario HabilitaUsuario DesabilitaUsuario ConectaServidor
Configuração	Configuração Estereoscópica - AtivaEstereo - DestivaEstereo - AumentaDistanciaCameras - DiminuiDistanciaCameras - AumentaHIT - DiminuiHIT Configuração de Rastreamento do Observador - AtivaRastreamentoObservador - DesativaRastreamentoObservador Configuração de Rastreamento da Vara de Seleção - CriaVaraSelecao - AumentaVaraSelecao - DiminuiVaraSelecao - RemoveVaraSelecao

ANEXO B

TIPOS DE EVENTO DO SISTEMA FADAS

1 Eventos do Mouse

1.1 Botão Mouse Pressionado

Este evento é gerado sempre que um botão do mouse é pressionado. Um parâmetro do tipo número deve ser declarado na rotina de tratamento para receber o código do botão.

1.2 Botão Mouse Liberado

Este evento é gerado sempre que um botão do mouse é liberado. Um parâmetro do tipo número deve ser declarado na rotina de tratamento do evento para receber o código do botão.

2 Eventos do Teclado

2.1 Tecla Pressionada

Este evento é gerado sempre que uma tecla é pressionada. Um parâmetro do tipo número deve ser declarado na rotina de tratamento do evento para receber o código da tecla.

2.2 Tecla Liberada

Este evento é gerado sempre que uma tecla é liberada. Um parâmetro do tipo número deve ser declarado na rotina de tratamento do evento para receber o código da tecla.

3 Eventos do Joystick

3.1 Botão Mouse Pressionado

Este evento é gerado sempre que um botão do joystick é pressionado. Um parâmetro do tipo número deve ser declarado na rotina de tratamento para receber o código do botão.

3.2 Botão Mouse Liberado

Este evento é gerado sempre que um botão do joystick é liberado. Um parâmetro do tipo número deve ser declarado na rotina de tratamento do evento para receber o código do botão.

4 Eventos de objeto tridimensional

4.1 Colisão Objeto

Este evento é gerado sempre que um objeto em movimento colide com outro objeto. Dois parâmetros do tipo texto devem ser declarados na rotina de tratamento do evento. O primeiro parâmetro recebe o nome do objeto em movimento e o segundo parâmetro recebe o nome do objeto colidido.

CAPÍTULO 7

LISTA DE REFERÊNCIAS

- ALICE. Pittsburgh. /2003/. Apresenta Alice Free, Easy, Interactive 3D Graphics for the WWW. Carnegie Mellon University. Disponível em: <<http://www.alice.org>>. Acesso em: 30 de set. 2003.
- AKKA, B. Writing Stereoscopic Software for StereoGraphics systems using Microsoft® Windows OpenGL. 1998. San Rafael. StereoGraphics Corporation. Disponível em: <<http://www.stereographics.com/pcsdk.htm>>. Acesso em: 09 nov. 1999.
- ALLISON, D.; HODGES, L.F. Virtual reality for education ? In: ACM SYMPOSIUM ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, Seoul, 2000. *Proceedings*. New York: ACM Press, 2000. p.160-165.
- ANDERSEN, H.C. *Contos de Andersen*. Trad. de Guttorm Hanssen. Rio de Janeiro: Paz e Terra, 1991.
- ASCENSION TECHNOLOGY CORPORATION. Burlington. /2003/. Apresenta o produto Flock of Birds. Ascension Technology Corporation. Disponível em: <<http://www.ascension-tech.com/products/flockofbirds.php>>. Acesso em: 30/09/2003.
- BATTAIOLA, A.L. et al. Desenvolvimento de um software educacional com base em conceitos de jogos de computador. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 13., São Leopoldo, 2002. *SBIE 2002: Metodologias, tecnologias e aprendizagem dentro do cenário de informática na educação*. São Leopoldo: UNISINOS, 2002. p.282-290.
- CASTRO, E.M.M. *O fim visual do século XX e outros textos críticos*. São Paulo, Editora da Universidade de São Paulo, 1993.
- CONWAY, M.J.; PAUSCH, R. Alice: easy to learn interactive 3D graphics. ACM SIGGRAPH Computer Graphics, New York, v.31, n.3, p.58-59, 1997.
- CONWAY, M.J. Alice: easy-to-learn 3D scripting for novices. 1997. 242p. Dissertation (Doctor of Philosophy) – School of Engineering and Applied Science, University of Virginia. Disponível em Rascunho de dissertação (Doutorado). 1998. Disponível em: <<http://www.alice.org/publications/pubs.html>>. Acesso em: 30 set. 2003.

CRUZ-NEIRA, C. et al. The CAVE: audio visual experience automatic virtual environment. Communications of the ACM, New York, v.35, n.6, p.64-72, 1992. New York

CRUZ-NEIRA, C.; SANDIN, D.J.; DEFANTI, T.A. Surround-screen projection-based virtual reality: the design and implementation of the CAVE. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 20., 1993. *Proceedings*. New York: ACM Press, 1993. p.135-142.

CRUZ-NEIRA, C. Introduction to virtual reality. In: SIGGRAPH 97, course 15 - applied virtual reality, Los Angeles, 1997.

DICKERSON, J.A. CRCDC: Low-Power wireless communications for virtual environments. In: AMERICAN SOCIETY FOR ENGINEERING EDUCATION ANNUAL CONFERENCE CONFERENCE, Toronto, 2002. *Proceedings*. Disponível em: <<http://clue.eng.iastate.edu/~julied/publications.html>>. Acesso em: 29 set. 2003.

DRUIN, A. (Ed.). *The design of children's technology*. San Francisco: Morgan Kaufmann Publishers, Inc., 1999.

FOLEY, J.D. et al. *Computer Graphics: principles and practice*. 2.ed. Addison Wesley, 1990.

FREINET, C. *Pedagogia do bom senso*. Trad. de J. Baptista. 2.ed. São Paulo: Martins Fontes, 1988.

FREIRE, P. *Pedagogia da indignação: cartas pedagógicas e outros escritos*. São Paulo: UNESP, 2000.

GARDNER, H. *Inteligências Múltiplas: a teoria na prática*. Trad. de Maria Adriana Veríssimo Veronese. Porto Alegre: Artes Médicas, 1995.

GAVA, T.B.S.; MENEZES, C.S. Moonline: proposta de um ambiente baseado na Web para apoio à construção coletiva de conhecimento. *Revista Brasileira de Informática na Educação*, Porto Alegre, v.10, n.2, p.21-35, 2002.

GÓES, L.P. *Pretinha e Branquinha*. Aparecida: Editora Santuário, 1991.

GÓES, L.P. *A aventura da literatura para crianças*. 5.ed. São Paulo: Melhoramentos, 1993.

GÓES, L.P. *Olhar de descoberta*. São Paulo: Edições Paulina, 2003.

GÓES, V.P.S.; ZUFFO, M.K.; LOPES, R.D. FADAS: um sistema para criação e manipulação de mundos virtuais em ambiente imersivo tridimensional. In: SYMPOSIUM ON VIRTUAL REALITY, 6., Ribeirão Preto, 2003. SVR2003.

GRUBER D. The Mathematics of the 3D rotation matrix. In: XTREME GAME DEVELOPERS CONFERENCE, 2000. Santa Clara, 2000. Disponível em: <<http://www.makegames.com/3Drotation>>. Acesso em: 30 set. 2003.

GUZDIAL, M.; SOLOWAY, E. Teaching the Nintendo Generation to Program. *Communications of the ACM*, New York, v.45, n.4, p.17-21, Apr. 2002.

HAWKINS, K.; ASTLE, D. *OpenGL Game Programming*. Roseville: Prima Publishing, 2001.

IGARASHI, T.; MATSUOKA, S.; TANAKA, H. Teddy: A Sketching Interface for 3D Freeform Desing. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 26., 1999. *Proceedings*. New York: ACM Press/Addison-Wesley Publishing, 1999. p.409-416.

JACKSON, R.L.; FAGAN, E. Collaboration and learning within immersive virtual reality. In: INTERNATIONAL CONFERENCE ON COLLABORATIVE VIRTUAL ENVIRONMENTS, 3., San Francisco, 2000. *Proceedings*. New York: ACM Press, 2000. p.83-92.

JARDIN, A. *Cibermãe: uma viagem extraordinária dentro do computador*. Trad. de Estela dos Santos Abreu. São Paulo: Moderna, 1998.

JONHNSON, A. et al. The NICE Project: learning together in a virtual world. In: VIRTUAL REALITY ANNUAL INTERNATIONAL SYMPOSIUM, Atlanta, 1998. *Proceedings*. 1998. p.176-183.

JOSÉ NETO, J. *Introdução à compilação*. Rio de Janeiro: LTC – Livros Técnicos e Científicos Editora, 1987.

KISHIMOTO, T.M. *Escolarização e brincadeira na educação infantil*. Texto teórico do Laboratório de Brinquedos e Materiais Pedagógicos (LABRIMP). Faculdade de Educação (FEUSP). São Paulo, /1996/. Disponível em: <<http://www.fe.usp.br/laboratorios/larimb/escola.htm>>. Acesso em: 26 set. 2003.

LASSETER, J. Principles of traditional animation applied to 3D computer animation. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 14., 1987. *Proceedings*. New York: ACM Press. p.34-44.

LÉVY, P. *Educação e Cybercultura*. Artigo. Faculdade de Educação da Universidade Federal de Minas Gerais (FaE/UFMG). Belo Horizonte, /1995/. Disponível em: <<http://www.fae.ufmg.br/catedra/artigo3.htm>>. Acesso em: 26 set. 2003

MCMANUS, M.M. Computer supported collaborative learning. *ACM SIGGROUP Bulletin*, New York, v.18, n.1, p.7-9, 1997.

MARCOTTY, M.; LEDGARD, H. *Programming language landscape: syntax, semantics, implementation*. 2.ed. Chicago: SRA, 1986.

MARINHO, S.P. Tecnologia, educação contemporânea e desafios ao professor. In: JOLY, M.C.R.A. (Org.). *A Tecnologia no ensino: implicações para a aprendizagem*. São Paulo: Casa do Psicólogo, 2002. p.41-62.

NAEF, M.; STAADT, O.; GROSS, M. Spatialized audio rendering for immersive virtual environments. In: ACM SYMPOSIUM ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, Hong Kong, 2002. *Proceedings*. New York: ACM Press, 2002. p.65-72.

NETTO, A.V; MACHADO, L.S.; OLIVEIRA, M.C.F. *Realidade Virtual: definições, dispositivos e aplicações*. Notas Didáticas, n. 34. ICMC-USP, São Carlos, 2002. 33p.

NIED. Núcleo de Informática Aplicada à Educação. *Projeto EDUCOM: Proposta Original*. Campinas: NIED, 1983. (Memo n.1).

NIED. Núcleo de Informática Aplicada à Educação. *EDUCOM: Avaliação dos Resultados obtidos no 1º ano de Implantação*. Campinas: NIED, 1986. (Memo n.2).

NIED. Núcleo de Informática Aplicada à Educação. *Criação de ambiente Logo para estudo do processo de alfabetização numa escola pública*. Campinas: NIED, 1987. (Memo n.14).

NIED. Núcleo de Informática Aplicada à Educação. *O uso de Logo em biologia: Modismo ou Necessidade ?*. Campinas: NIED, 1988. (Memo n.21).

NIED. Núcleo de Informática Aplicada à Educação. *Introdução à programação Logo*. Campinas: NIED, 1990. (Memo n.21).

NOMA, H.; SUGIHARA, T.; MIYASATO, T. Development of Ground Surface Simulator for Tel-E-Merge System. In: IEEE VIRTUAL REALITY 2000 CONFERENCE, 2000. *Proceedings*. p.217-224. Disponível em: <<http://www.mis.atr.co.jp/~noma/papers/index.e.html>>. Acesso em: 27/09/2003.

PAPERT, S. *Does easy do it ? Children, games, and learning*. Game Developer magazine, s.l., p.88, June 1998. Disponível em: <<http://www.papert.org/articles/doeseasydoiti.html>>. Acesso em: 26 set. 2003.

PAPERT, S. Introduction: What is Logo ? Who need it ?. In: LCSi (Logo Computer System Inc.). *Logo philosophy and implementation*. USA: LCSi, 1999a. p.5-16.

PAPERT, S. *Papert on Piaget*. Time magazine, special issue on “The Century’s Greatest Minds”, p.105, March 29, 1999b. Disponível em: <<http://www.papert.org/articles/papertonpiaget.html>>. Acesso em: 26 set. 2003.

PINHO, M.S. Realidade Virtual como ferramenta de informática na educação. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA, 7., Belo Horizonte, 1996. Publicações do Grupo de Realidade Virtual da PUCRS. Resumo. Porto Alegre: PUCRS, 1996. Disponível em: <<http://grv.inf.pucrs/pagina/educa/educa.htm>>. Acesso em: 27 set. 2003.

PINHO, M.S. et al. A User Interface Model for Navigation in Virtual Worlds. In: CONFERÊNCIA LATINO-AMERICANA DE INFORMÁTICA, Assunción, 1999. Publicações do Grupo de Realidade Virtual da PUCRS. Porto Alegre: PUCRS, 1999. Disponível em: <<http://grv.inf.pucrs.br/pagina/publicacoes/publicacoes.htm>>. Acesso em: 27 set. 2003.

PINHO, M.S. *Técnicas de interação em ambientes tridimensionais*. In: WORKSHOP DE REALIDADE VIRTUAL, Gramado, 2000. Publicações do Grupo de Realidade Virtual da PUCRS. Porto Alegre: PUCRS, 2000. Disponível em: <<http://grv.inf.pucrs.br/Pagina/Publicacoes/Publicacoes.htm>>. Acesso em: 27 set. 2003.

PINHO, M.S.; FREITAS, C.M.D. Cooperative and simultaneous object manipulation in collaborative virtual environments. In: IEEE WORKSHOP ON 'THE FUTURE OF VR AND AR INTERFACES', 2001. *VR2001*. Yopkohama, 2001.

PROINFO. Apresenta o programa educacional. Brasília: MEC, /1997/. Disponível em: <<http://www.proinfo.mec.gov.br>>. Acesso em 24 set. 2003.

ROUSSOS, M. et al. Learning and Building Together in an Immersive Virtual World. In: PRESENCE, 3., 1999. *Virtual Environments and Learning*. Massachusetts: MIT Press, 1999. p.247-263.

SANTAELLA, L. *Cultura das mídias*. São Paulo: Experimento, 1996.

SANTOS, E.T. et al. Computação gráfica: estado da arte e a pesquisa na USP. In: SIMPÓSIO NACIONAL DE GEOMETRIA DESCRITIVA E DESENHO TÉCNICO, 15., São Paulo, 2001. *GRAPHICA 2001*. São Paulo: EPUSP, 2001. p. 3.

SOARES, L.P. et al. Powering multiprojection immersive environments with cluster of commodity computers. In: IBERO-AMERICAN SYMPOSIUM IN COMPUTER GRAPHICS, 1., Guimarães, 2002. *SIACG 2002*. Guimarães: 2002.

STAGE 3. Pittsburgh. /2003/. Apresenta Stage 3 Research Group. Carnegie Mellon University. Disponível em <<http://www.alice.org/stage3>>. Acesso em 30 set. 2003.

STEREOGRAPHICS CORPORATION. StereoGraphics developers' handbook: background on creating images for CrystalEyes and SimulEyes. San Rafael. 1997.

StereoGraphics Corporation. Disponível em: <<http://www.stereographics.com>>. Acesso em: 09 nov. 1999.

SUTHERS, D.D. Architectures for Computer Supported Collaborative Learning. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 2001. *ICALT2001*. 2001. p.25-28.

THALMANN, N.M.; THALMANN, D. *Computer animation in future technologies*. Geneva. 1998. Texto. MIRAlab – University of Geneva. Disponível em: <www.miralab.unige.ch/papers/105.pdf>. Acesso em: 28 set. 2003.

VALENTE, J.A. *Por quê o computador na educação*. Texto do PROINFO. Brasília: MEC, /1993/. Disponível em: <<http://www.proinfo.mec.gov.br/biblioteca/textos/>>. Acesso em: 24 set. 2003.

VALENTE, J.A.; ALMEIDA, F.J. *Visão Analítica da Informática na Educação no Brasil*. Texto do PROINFO. Brasília: MEC, /1993/. Disponível em: <<http://www.proinfo.mec.gov.br/biblioteca/textos/>>. Acesso em: 24 set. 2003.

VALENTE, J.A.(Org.) Informática na educação no Brasil: análise e contextualização histórica. In: VALENTE, J.A. *O computador na sociedade do conhecimento*. Campinas: UNICAMP/NIED, 1999. p1-27.

VALENTE, J.A. Aprendendo para a vida: o uso da informática na educação especial. In: FREIRE, F.M.P.; VALENTE, J.A.(Orgs.). *Aprendendo para a vida: os computadores na sala de aula*. São Paulo: Cortez, 2001. p.29-42.

VIEIRA, F.M.S. *A utilização das novas tecnologias na educação numa perspectiva construtivista*. Texto do PROINFO. Brasília: MEC, /1998/. Disponível em: <<http://www.proinfo.mec.gov.br/biblioteca/textos/>>. Acesso em: 24 set. 2003.

WAZLAWICK, R.S. et al. Collaborative learning in a virtual museum authoring tool. In: SYMPOSIUM ON VIRTUAL REALITY, 2001. Florianópolis, 2001.

WESTON, D.C.; WESTON, M.S. *Aprender Brincando: atividades divertidas para construir o caráter, a consciência e a inteligência emocional das crianças*. Trad. de Marcos Vianna Van Acker. São Paulo: Paulinas, 2000.

WHEATSTONE, C. *Contributions to the physiology of vision - part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision*. Londres: 1838. Artigo da biblioteca. Stereoscopy.com. Disponível em: <<http://www.stereoscopy.com/library/wheatstone-paper1838.html>>. Acesso em: 29 set. 2003.

WINN, W. The impact of three-dimensional immersive virtual environment on modern pedagogy. Seattle, 1997. Technical Report R-97-15. Human Interface Technology Laboratory (HITL) of University of Washington. Disponível em: <<http://www.hitl.washington.edu/publications/r-97-15>>. Acesso em: 27 set. 2003.

ZUFFO, M.K.; SOARES, L.P.; STEIN, A.; LOPES, R.D. CAVERNA Digital - um ambiente de multiprojeção para a imersão do usuário em realidades artificiais. In: CONGRESSO ARTE E CIÊNCIA, 4., São Paulo, 2001. *Anais: mito e razão*. São Paulo: Ed. Elza Ajzenberg, 2001. p.45-54.

ZUFFO, J.A. *A sociedade e a economia no novo milênio: os empregos e as empresas no turbulento alvorecer do Século XXI*, livro 1: a tecnologia e a infossociedade. Barueri: Malone, 2003.

APÊNDICE I

GRAMÁTICA PARA RECONHECIMENTO LÉXICO DA LINGUAGEM FADAS

A seguir se descreve a gramática para reconhecimento léxico dos átomos da linguagem Fadas, através da notação de Wirth.

Átomo

$A = N \mid R \mid T \mid E \mid S$

Número

$N = [+|-]D\{D\}[D\{D\}]$.

$D = 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$.

Sinal

$S = : \mid ; \mid p \mid \# \mid + \mid - \mid * \mid / \mid < \mid > \mid j \mid k \mid d \mid m \mid m$.

Cor

$R = (C|c) < D\{D\}; D\{D\}; D\{D\} >$.

Texto

$T = \{L|D|_ \}$.

$L = A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|X|Y|Z$.

$|a|b|c| \dots z$

$|Á|É|Í|Ó|Ú|Ç|Ã|Õ|Û|Ê|Ô$

$|á|é|í|ó|ú|ç|ã|õ|ü|ê|ô$.

Identificador

$I = L\{L|D|_ \}$.

$L = a|b|c|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|x|y|w|z$.

Vetor

$E = (C|c) < D\{D\}; D\{D\}; D\{D\} >$.

Observação

Dois átomos numéricos consecutivos não são permitidos.

Legenda de terminais do tipo sinal

p – ponto final.
; - separador de parâmetros
* - multiplicação
< - abertura de parênteses
> - fechamento de parênteses
- sinal de igual
j – sinal de comparação maior
k – sinal de comparação menor
d - sinal de comparação diferente
m - sinal de comparação maior ou igual
n - sinal de comparação menor ou igual.

APÊNDICE II

GRAMÁTICA DOS COMANDOS BÁSICOS DA LINGUAGEM FADAS

A seguir se descreve os comandos básicos da linguagem Fadas através da notação de Wirth.

Comando

$C = (A|Ç)p.$

Atribuição de Característica

$A = i : i[:i] \# Fp.$

Ação

$Ç = i : i[:i] [L]p.$

$L = \{ F; \} F.$ (L = lista de parâmetros)

Valor de informação

$F = n | c | t | v.$

Legenda de terminais

p – sinal de ponto final.

i - identificador

- sinal de igual.

t – texto.

n – número.

c – cor.

v – vetor.

APÊNDICE III

GRAMÁTICA DA LINGUAGEM FADAS

$P = oS\{S\}fo.$	Programa Fadas
$S = C D.$	Sentença da linguagem Fadas.
$C = Ç A I B G J K H.$	Comando.
$D = X R V.$	Declaração.
$Ç = i:i[:i][L]p.$	Ação
$L = \{M;\}M.$	Lista de parâmetros de uma ação ou rotina.
$M = F i E T.$	Parâmetro de rotina.
$T = i:i[:i].$	Característica
$F = n c t v.$	Valor de informação.
$A = i:i[:i]\#Mp.$	Atribuição de valor para uma característica
$I = i\#Mp.$	Atribuição de valor para uma caixa
$E = O[*O/O][+O[*O/O]]-O[*O/O].$	Expressão numérica
$O = n i T <E>.$	Operando
$B = sPã.$	Abertura de condição
$P = O(\# d j k m n)O.$	Comparação numérica
$Q = e.$	Alternativa de condição (senão)
$G = fsp.$	Fechamento de condição
$J = qP.$	Abertura de laço
$K = fqp.$	Fechamento de laço
$H = i[L]p.$	Chamada de rotina
$X = rxip.$	Declaração de uma caixa.
$R = rai[L].$	Abertura de uma declaração de uma rotina.
$U = fap.$	Fechamento de declaração de uma rotina.
$V = rbit;tp.$	Declaração de evento.

Legenda de terminais

p – sinal de ponto final.
i – identificador.
- sinal de igual.
t – texto.
n – número.
c – cor.
v – vetor.
< - abertura de parêntese
> - fechamento de parêntese
- sinal de igual
j – sinal de comparação maior
k – sinal de comparação menor
d - sinal de comparação diferente
m - sinal de comparação maior ou igual
n - sinal de comparação menor ou igual.
o – palavra reservada Programa.
f – palavra reservada Fim.
s – palavra reservada Se.
ã – palavra reservada Então.
e – palavra reservada Senão
q – palavra reservada Enquanto
r – palavra reservada Cria
x – palavra reservada Caixa
a – palavra reservada Rotina
b – palavra reservada Evento

APÊNDICE IV

Lista de Mensagens de Erro do Sistema Fadas

1 Mensagens do analisador léxico

- a) Cadeia de caracteres inválida

2 Mensagens do analisador sintático

- a) Sentença não reconhecida.
- b) Sentença de final de programa não encontrada.
- c) Sentença de final de condição não encontrada.
- d) Sentença alternativa de condição sem início de condição.
- e) Sentença de final de laço não encontrada.
- f) Sentença de final de rotina não encontrada.

3 Mensagens do analisador semântico

- a) Caixa já declarada anteriormente.
- b) Caixa não declarada.
- c) Rotina já declarada anteriormente.
- d) Rotina não declarada.
- e) Evento já declarado anteriormente.
- f) Parâmetros de ação incorretos.
- g) Parâmetros de chamada de rotina incorretos.
- h) Objeto não encontrado.
- i) Ação inválida para o objeto.
- j) Característica inválida para o objeto.
- k) Valor inválido para a característica do objeto.
- l) Caixa com valor inválido para expressão numérica.

- m) Característica com valor inválido para expressão numérica.
- n) Cor inválida.
- o) Tipo de evento inválido.

4 Mensagens específicas de ações e características de objetos

- a) Direção inválida.
- b) Ângulo inválido.
- c) Parâmetro com valor inválido.
- d) Nome já utilizado por outro objeto.
- e) Ângulo inválido.
- f) Animação não encontrada.
- g) Falha na criação do fonte de texto tridimensional.
- h) Nenhum vértice encontrado.
- i) Arquivo não encontrado.
- j) Formato de arquivo não suportado.
- k) Endereço IP inválido.
- l) Usuário não encontrado.

5 Mensagens de configuração de hardware e software

- a) Geração de imagem estereoscópica não suportada.
- b) Cronômetro de hardware não encontrado.
- c) Falha ao iniciar o sistema de som.
- d) Falha na conexão com o servidor Fadas.
- e) Falha na conexão do console com a aplicação Fadas.

APÊNDICE V

CONFIGURAÇÃO DO SISTEMA FADAS

1 Arquivo de configuração de aplicação

Este é um arquivo texto (código ASCII) com o nome FADAS.INI localizado no mesmo diretório dos executáveis FADAS.EXE e CONSOLEFADAS.EXE. A sua estrutura e manipulação segue o padrão de arquivos de configuração (“initialization file”) da plataforma Windows.

Na seção [IP] o endereço IP e porta são configurados para a aplicação gráfica Fadas e aplicação console.

Na seção [Geral], a chave MsgUsu indica se mensagens devem ser apresentadas na aplicação gráfica, ou somente na aplicação console. A chave DirFadas define o diretório de instalação do sistema e a chave Estereo indica se imagens estereoscópicas devem ser geradas. Os indicadores podem receber os valores zero ou um, sendo que o valor um ativa o indicador e o zero desativa-o.

A seção [Mestre] é utilizada para instalação em ambiente de multi-projeção e define para a aplicação mestre o endereço IP e porta para das aplicações clientes, responsáveis por diferentes projeções.

A seção [Cliente] é utilizada no computador cliente de uma instalação em ambiente de multi-projeção. Ela define o identificador do cliente, a face de projeção e o endereço IP e porta da aplicação mestre.

A seção [Servidor] é utilizada para instalação em ambiente colaborativo. Ela define o endereço IP e porta de cada usuário, o seu apelido e um indicador de habilitação.

A seção [Usuário] é utilizada no computador cliente de uma instalação em ambiente colaborativo. Ela define o endereço IP e porta do servidor e o apelido do usuário.

A seguir é apresentado um exemplo de arquivo de configuração Fadas.

Exemplo:

```
; Arquivo de configuração da aplicação Fadas
; Aplicação Única (standalone)
```

```
[IP]
```

```
ipGraf=192.168.0.31
```

```
ptaGraf=1003
```

```
ipConsole=192.168.0.31
```

```
ptaConsole=1004
```

```
[Geral]
```

```
MsgUsu=1
```

```
DirFadas=C:\Fadas
```

```
Estereo=0
```

```
:[Mestre]
```

```
;ipCliente1=192.168.0.31
```

```
;ptaCliente1=1005
```

```
:[Cliente]
```

```
;IDCliente=0
```

```
;Face=Direita
```

```
;ipMes=192.168.0.31
```

```
;ptaMes=1003
```

```
:[Servidor]
```

```
;ipUsuario1 = 192.169.0.31
```

```
;ptaUsuario1 =1005
```

```
;Apelido1=Junior
```

```
;Habilitado1=1
```

```
;ipUsuario2 = 192.169.0.31
```

```
;ptaUsuario2 =1006
```

```
;Apelido2=Pedro
```

```
;Habilitado2=0
```

```
:[Usuario]
```

```
;ipServidor = 192.169.0.31
```

```
;ptaServidor =1003
```

```
;Apelido=Junior
```

2 Opções de linha de comando

As configurações de vídeo são definidas na linha de comando da aplicação Fadas através das opções TelaCheia (1 para ativar tela cheia), LarguraTela (pixels da largura da tela) e AlturaTela (pixels da altura da tela).

Exemplo: ativa o aplicativo gráfico Fadas com tela cheia 1024 x 768 pixels.

```
C:\Fadas\Fadas.exe TelaCheia=1 LarguraTela=1024 AlturaTela=768
```