

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SISTEMA DE AUTORIA PARA CONSTRUÇÃO DE
“ADVENTURES” EDUCACIONAIS EM REALIDADE VIRTUAL

Patrícia Cristiane de Souza

Florianópolis, fevereiro de 1997

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E DE ESTATÍSTICA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
ÁREA DE CONCENTRAÇÃO: SISTEMAS DE CONHECIMENTO

SISTEMA DE AUTORIA PARA CONSTRUÇÃO DE
“ADVENTURES” EDUCACIONAIS EM REALIDADE VIRTUAL

por Patrícia Cristiane de Souza

Orientador: Prof. Raul Sidnei Wazlawick, Dr.

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina, como parte dos requisitos exigidos para obtenção do grau de Mestre em Ciência da Computação.

Florianópolis, fevereiro de 1997

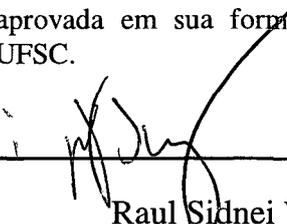
***Sistema de Autoria para Construção de “Adventures”
Educativas em Realidade Virtual***

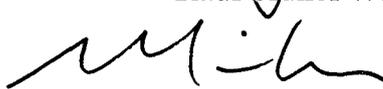
Patrícia Cristiane de Souza

Esta dissertação foi julgada adequada para a obtenção do título de

MESTRE EM CIÊNCIA DA COMPUTAÇÃO

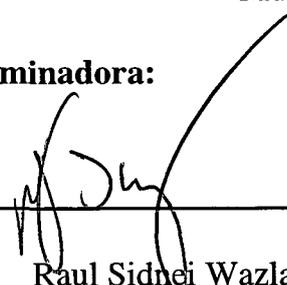
na área de concentração Sistemas de Conhecimento e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação da UFSC.



Raul Sidnei Wazlawick


Murilo Silva de Camargo - Coordenador do CPGCC/UFSC

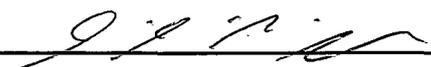
Banca Examinadora:



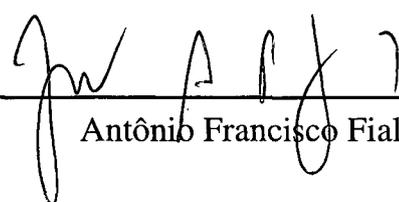
Raul Sidnei Wazlawick



Edla Maria Faust Ramos



Aginaldo Robson de Souza



Antônio Francisco Fialho

Não há nada mais difícil de se realizar, nem cujo êxito seja mais duvidoso nem cuja execução seja mais perigosa, do que iniciar uma nova ordem das coisas. Pois o reformista tem como inimigos todos aqueles que lucram com a antiga ordem e tem como defensores não muito entusiastas apenas aqueles que lucram com a nova ordem, sendo essa falta de entusiasmo proveniente em parte do temor aos adversários, que têm as leis a seu favor, e em parte da incredulidade da Humanidade, que não acredita realmente em nada que ainda não tenha experimentado.

(Niccollo Machiavelli)

AGRADECIMENTOS

Ao CNPq órgão financiador deste projeto.

A coordenação e a secretaria do programa de pós-graduação, pela presteza no atendimento, paciência e carinho.

Ao prof. Raul S. Wazlawick, pelo voto de confiança, pela amizade, e principalmente pela orientação construtivista, oportunizando-me a construção de novos conhecimentos.

Aos professores integrantes da banca examinadora pelas sugestões, críticas, ou seja, pela valorização que suas contribuições prestaram ao presente trabalho.

Ao Augusto Hoffmann pela participação ativa no projeto durante os seis meses finais.

A irmã arquiteta Giselle, pela maciça contribuição sobre as áreas de engenharia e arquitetura.

Aos amigos de pós-graduação e aos de toda hora - Cristiano, Karithia, Ana Carolina, Soraia, Ricardo, Luciana e tantos outros pela confiança, entusiasmo, e principalmente pelo ombro.

E finalmente, a todos familiares, pelo amor, pelas orações e principalmente pela unidade.

RESUMO

O presente trabalho envolve a aplicação da tecnologia computacional à educação. Discute-se a importância pedagógica da utilização de metáforas altamente interativas na construção de *software* educacional e apresenta-se um modelo de interação utilizando jogos educacionais em realidade virtual.

A crença de que o aprendiz constrói seu conhecimento quando ele é parte integrante de um universo e interage diretamente com ele, torna-o um agente ativo do seu processo de aprendizado. A partir desta crença, buscou-se especificar uma ferramenta de autoria para a criação de ambientes educacionais baseados na metáfora dos “*Role-Playing Games*” ou também conhecidos como “*Adventure Games*”, com o intuito de construir “*adventures educacionais*” que se apresentassem como bons ambientes de aprendizado.

Tal ferramenta, pretende ser também um facilitador à criação de ambientes educacionais, a ponto de que o próprio professor ou até mesmo os alunos tornem-se autores de jogos educativos.

Um protótipo da ferramenta de autoria, foi especificado e em parte desenvolvido, bem como a criação de um ambiente prototipado de uma possível aplicação desta ferramenta para demonstração de suas potencialidades. Este ambiente relaciona conceitos relativos à aprendizagem de engenharia civil.

ABSTRACT

This work is about the application of computational technology in education. It discusses the pedagogic significance of using highly interactive metaphors in the construction of educational software and it introduces an interaction model using educational games in virtual reality.

The belief that the apprentice himself builds knowledge when he takes part of an environment and interacts with it makes him an active agent of his own learning process. It was specified an authoring system for creating educational environments based in role-playing games and adventure games. It was the intention of this work to examine “educational adventures” that are thought to be good learning environments.

The authoring system is intended to make easy for the teachers creating educational environments.

An authoring system prototype was specified and partially developed. Also, a prototype model of an application was developed in order to show the system’s capabilities. This application includes concepts related to engineering.

PALAVRAS-CHAVE

adventure games

realidade virtual

role-playing games

sistema de autoria

software educacional

KEYWORDS

adventure games

authoring system

educational software

role-playing games

virtual reality

SUMÁRIO

AGRADECIMENTOS	V
RESUMO	VI
ABSTRACT	VII
PALAVRAS-CHAVE.....	VIII
KEYWORDS	IX
SUMÁRIO	X
LISTA DE FIGURAS.....	XIV
LISTA DE TABELAS	XV
LISTA DE ABREVIACÕES	XVI
1. INTRODUÇÃO	1
1.1. APRESENTAÇÃO.....	1
1.2. DEFINIÇÃO DO PROBLEMA	2
1.3. OBJETIVOS.....	3
1.4. ESTRUTURA.....	3
2. FUNDAMENTAÇÃO TEÓRICA.....	5
2.1. INTRODUÇÃO.....	5
2.2. O ESTUDO DE JEAN PIAGET	5
2.2.1. <i>A Teoria da Equilibração Cognitiva</i>	5
2.2.2. <i>Os Períodos da Construção da Inteligência</i>	7
2.2.2.1. Estágio Sensório-Motor (0 - 2 anos)	7
2.2.2.2. Estágio Pré-Operatório (2 à 7 anos)	8
2.2.2.3. Estágio Operatório Concreto (7 à 11 anos)	8
2.2.2.4. Estágio Operatório Formal (a partir dos 12 anos)	9
2.2.3. <i>Comentários acerca da Teoria Construtivista</i>	9
2.3. APRENDIZAGEM ATRAVÉS DA CONSTRUÇÃO DO CONHECIMENTO	10
3. FERRAMENTAS DE AUTORIA	12
3.1. INTRODUÇÃO.....	12
3.2. MULTIMÍDIA TOOLBOOK	13
3.2.1. <i>Histórico</i>	13

3.2.2. <i>Características Gerais</i>	14
3.2.3. <i>Multimídia Toolbook 4.0 (MTB 4.0)</i>	15
4. SOFTWARE EDUCACIONAL	17
4.1. INTRODUÇÃO.....	17
4.2. JOGOS.....	17
4.2.1. <i>Jogos via Internet</i>	17
4.2.1.1. MUDs.....	18
4.2.1.2. Clientes.....	19
4.2.1.3. Servidores.....	20
4.2.2. <i>Jogos Monousuários</i>	21
4.2.2.1. TIM - The Incredible Machine.....	21
4.2.2.2. SIMCITY.....	22
4.3. AMBIENTES EDUCACIONAIS.....	23
4.3.1. <i>O Projeto EPES</i>	24
4.3.1.1. Software 'ALICE'.....	24
4.3.1.2. Software 'A ALDEIA VIKING'.....	26
4.3.2. <i>Software Rocky's Boots e Software Robot Odyssey</i>	28
5. REALIDADE VIRTUAL	30
5.1. CLASSES DE SISTEMAS DE REALIDADE VIRTUAL.....	32
5.2. DISPOSITIVOS DE REALIDADE VIRTUAL.....	33
5.2.1. <i>Dispositivos de Visão</i>	34
5.2.1.1. Stereo Glasses ou Shutter Glasses.....	34
5.2.1.2. Head Mounted Displays.....	34
5.2.2. <i>Dispositivos de Rastreamento</i>	35
5.2.3. <i>Luvas Eletrônicas</i>	35
5.2.4. <i>Dispositivos Geradores de Sensação de Tato e de Força</i>	36
5.3. ASPECTOS DE UM PROGRAMA DE RV.....	36
5.3.1. <i>O Processo de Entrada</i>	36
5.3.2. <i>O Processo de Simulação</i>	36
5.3.3. <i>O Processo de Rendering</i>	37
5.3.4. <i>A Base de dados do mundo virtual</i>	37
5.3.5. <i>O Mundo Virtual</i>	38
6. REALIDADE VIRTUAL E O ENSINO	39
6.1. COMO, QUANDO E POR QUE USAR A RV NA EDUCAÇÃO.....	39
6.2. EXPERIÊNCIAS DO USO DE REALIDADE VIRTUAL APLICADA À EDUCAÇÃO.....	41
6.2.1. <i>ExploreNet</i>	42
6.2.2. <i>VETL - Virtual Environment Technology Laboratory</i>	42

6.2.3. <i>Educação Especial</i>	43
6.2.4. <i>VESAMOTEX - Virtual Education Science and Math of Texas</i>	44
6.2.5. <i>O Ensino de Literatura</i>	45
6.3. FONTES DE PESQUISA	46
7. SISTEMA DE AUTORIA	47
7.1. ESPECIFICAÇÃO DO SISTEMA.....	48
7.1.1. <i>Módulo Edição</i>	49
7.1.2. <i>Módulo Composição</i>	49
7.1.3. <i>Módulo de Interação</i>	50
7.1.4. <i>Bibliotecas</i>	53
7.1.5. <i>Sistema de Arquivos</i>	53
7.2. CONSTITUIÇÃO DOS “ADVENTURES EDUCACIONAIS”	54
7.2.1. <i>Características dos Micro-Mundos</i>	55
7.2.2. <i>Sistema de Arquivos dos Micro-Mundos</i>	56
8. APRESENTAÇÃO DA FERRAMENTA	57
9. ÁREA DA APLICAÇÃO	66
9.1. PRINCÍPIOS GERAIS DO PROJETO ESTRUTURAL.....	66
9.1.1. <i>Peças Estruturais</i>	66
9.1.2. <i>Análise da Estrutura</i>	67
9.1.3. <i>Fissuras</i>	69
9.1.4. <i>Condições gerais</i>	70
10. APRESENTAÇÃO DA APLICAÇÃO	71
11. CONSIDERAÇÕES FINAIS	74
11.1. METÁFORA DOS JOGOS EDUCATIVOS	74
11.2. FERRAMENTA DE AUTORIA.....	75
11.3. APLICAÇÃO	76
11.4. LINHAS DE PESQUISA.....	76
12. REFERÊNCIAS BIBLIOGRÁFICAS	77
ANEXO A - LINGUAGENS UTILIZADAS	83
A1. A LINGUAGEM JAVA	83
A1.1. DESCRIÇÃO GERAL	83
A1.2. RELAÇÃO JAVA VERSUS WWW.....	84
A1.3. PLATAFORMAS.....	85
A1.4. FERRAMENTAS	85

A1.5. PESQUISANDO JAVA.....	85
A2. A LINGUAGEM VRML.....	85
A2.1. INTRODUÇÃO.....	85
A2.2. DEFINIÇÃO.....	86
A2.3. COMPARANDO VRML COM HTML.....	86
A2.4. FERRAMENTAS PARA UTILIZAÇÃO E CONSTRUÇÃO DE MUNDOS VRML.....	88
A2.5. FORMATO DE UM ARQUIVO EM VRML.....	88
A2.6. APLICAÇÕES.....	91

LISTA DE FIGURAS

FIGURA 4.1: TELA DO JOGO TIM.....	22
FIGURA 4.2: TELA INICIAL DO <i>SOFTWARE</i> ALICE.....	26
FIGURA 4.3: APRESENTAÇÃO DE UMA DAS TELAS INICIAIS DO <i>SOFTWARE</i> ALDEIA VIKING.....	27
FIGURA 7.1: INTER-RELACIONAMENTO ENTRE OS MÓDULOS E BIBLIOTECAS DO SISTEMA.....	49
FIGURA 7.2: ÁRVORE DE DIRETÓRIOS DA FERRAMENTA DE AUTORIA.....	54
FIGURA 7.3: FIGURA ILUSTRATIVA SOBRE A CONSTITUIÇÃO DOS MICRO-MUNDOS.....	54
FIGURA 8.1: TELA INICIAL DA FERRAMENTA.....	57
FIGURA 8.2: TELA PROPRIEDADES DO CENÁRIO.....	57
FIGURA 8.3: TELA PRINCIPAL DA FERRAMENTA DE AUTORIA.....	58
FIGURA 8.4: TELA PROPRIEDADES DO OBJETO.....	59
FIGURA 8.5 : BIBLIOTECA DE IMAGENS DOS OBJETOS.....	60
FIGURA 8.6 : TELA DADOS GERAIS DO OBJETO PORTA.....	60
FIGURA 8.7: TELA VARIÁVEIS DO OBJETO PORTA.....	61
FIGURA 8.8: TELA DO MÓDULO DE INTERAÇÃO - MODO VISUAL.....	62
FIGURA 8.9: TELA DE ELABORAÇÃO DO TESTE CONDICIONAL “SE-ENTÃO”.....	62
FIGURA 8.10: TELA DE DEFINIÇÃO PARA O COMPORTAMENTO DO BOTÃO “ENTÃO”.....	63
FIGURA 8.11: TELA DE DEFINIÇÃO PARA O COMPORTAMENTO DO BOTÃO “SENÃO”.....	64
FIGURA 8.12: TELA DO MÓDULO INTERAÇÃO - MODO <i>SCRIPT</i>	64
FIGURA 8.13: BIBLIOTECA DE PERSONAGENS.....	65
FIGURA 8.14: BARRA DE FERRAMENTAS DO MÓDULO DE COMPOSIÇÃO.....	65
FIGURA 9.1 (A): MONTAGEM DAS PEÇAS ESTRUTURAIS - IMAGEM CORTE.....	67
FIGURA 9.1 (B): MONTAGEM DAS PEÇAS ESTRUTURAIS - IMAGEM ISOMÉTRICA.....	67
FIGURA 10.1: TELA DE APRESENTAÇÃO.....	72
FIGURA 10.2: VISÃO INTERNA DO MICRO-MUNDO.....	73
FIGURA A2.1: EXEMPLO DE ARQUIVO VRML [AME 96].....	89
FIGURA A2.2: UM CHAPÉU.....	91

LISTA DE TABELAS

TABELA 2.1: EVOLUÇÃO DOS ESQUEMAS DE CONSERVAÇÃO.....	9
TABELA 4.1: LISTA DE COMANDOS RELACIONADOS AOS OBJETOS.....	19
TABELA 4.2: LISTA DE COMANDOS RELACIONADOS À <i>ROOM</i>	19

LISTA DE ABREVIACOES

BSD	Berkeley Software Distribution
CAD	Computer Aided Design
CBT	Computer Based Training
CGI	Common Gateway Interface
CMS	Course Management System
CRT	Cathode Ray Tube
DDE	Dynamic Data Exchange
DLL	Dynamic Link Library
EPES	European Pool of Educational Software
FAQ	Frequently Asked Questions
FTP	File Transfer Protocol
HMD	Head Mounted Displays
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
LCD	Liquid Cristal Display
MIDI	Musical Instrument Digital Interface
MOO	MUDs Oriented Objects
MUD	Multi User Dungeons
PC	Personal Computer
RPGs	Role Playing Games
RV	Realidade Virtual
SysV	System Five
TCP	Transmission Control Protocol

URL	Uniform Resource Locator
VE	Virtual Environment
VMS	Virtual Machine System
VOC	Voice, padrão de arquivos de som digitalizados da Creative Labs
VPC	Visores Presos à Cabeça
VRML	Virtual Reality Modeling Language
WAV	Wave, padrão de arquivos de som digitalizados
WinSock	Windows Sockets
WWW	World Wide Web
WYSIWYG	What You See Is What You Get

1. INTRODUÇÃO

1.1. APRESENTAÇÃO

É incontestável a utilização da informática em todas as atividades profissionais e sociais, e o fascínio que esta exerce nos jovens. Portanto, se uma das finalidades da escola é preparar a integração dos alunos na vida ativa, torna-se cada vez mais importante a utilização do computador como uma realidade no processo de ensino-aprendizagem.

Baseado nisto, resta saber qual a melhor forma de explorar esta tecnologia como uma ferramenta de auxílio na construção das estruturas cognitivas do aluno durante o aprendizado.

Em geral, muitos *softwares* educacionais refletem uma concepção inadequada do processo de ensino-aprendizagem: a de que ensinar é a mera transmissão de conteúdos, fazendo com que o aluno continue sendo um agente passivo como ainda ocorre no ensino tradicional.

Um fator essencial à concepção de *software* educacional é que este seja “interativo”, isto é, o usuário deve estar em plena comunicação com o sistema e vice-versa. O usuário pode interagir com o sistema por diversos meios, através de resolução de problemas, da análise de representações gráficas, da simulação e da participação ativa no próprio ambiente, como um agente ativo do sistema - e é esta a abordagem adotada, mas para isto o sistema deve contar com uma interface amigável.

Neste momento, pode-se questionar como se dá esta interface. A interface em si, define-se em ser a própria conexão do *software* com o sistema sensorio-motor do homem. É evidente que, para se construir uma interface amigável, deve-se ter um abrangente conhecimento sobre as duas partes que irão se conectar.

Conforme Coutaz, em [COU 90], “... aquele que concebe um sistema interativo deve elaborar, uma descrição o mais precisa possível do problema e dos processos cognitivos do usuário.”

O ambiente em si, também deve estimular o raciocínio (levando o aluno a pensar, não

apenas a memorizar), permitir ao aluno autonomia, considerando que este possa explorar o ambiente livremente, traçando um percurso conforme seu interesse e desenvolvimento no assunto.

Uma outra questão fundamental à criação de *software* educacional, é a necessidade de uma formalização teórica. Seguindo a abordagem construtivista (Jean Piaget), o aluno deve ser vislumbrado não como um mero repositório de conhecimento, mas como um agente construtor deste conhecimento. Ele deve ser capaz de descobrir os conhecimentos intrínsecos aos objetos que manipula e não receber simplesmente estes conhecimentos prontos.

1.2. DEFINIÇÃO DO PROBLEMA

A problemática em questão, reside no fato de existir poucos *software* educacionais que suprem as características desejáveis (como já apresentado na seção 1.1), para torná-los ambientes educacionais de boa qualidade.

O uso de técnicas e ferramentas de hipermídia na educação, teve um elevado crescimento nos últimos anos. De um modo geral, pode-se observar que, o *software* hipermídia, apesar da riqueza visual e sonora (recursos de som, imagem, animação, etc.) que comporta, faz com que o aprendiz interaja com o ambiente como se estivesse manuseando um livro eletrônico. Infelizmente, na maioria dos casos, é pouca ou quase nula a possibilidade do aprendiz intervir nesse ambiente, para criar ou interagir de forma ativa sobre o objeto de conhecimento. E, quando essa interação é permitida, consiste apenas na capacidade de adicionar novos textos e imagens ao sistema existente.

Diante disto, buscou-se um paradigma alternativo, onde o aprendiz interage ativamente na construção de conhecimento. Um exemplo de ambiente com tais características são os jogos do tipo “adventure”, também conhecidos como “role playing games”. Em tais jogos, os participantes são personagens de uma história que se desenrola, e suas ações vão modificar esta história à medida que ela acontece. Os participantes vêm-se frente a situações que exigem criatividade e conhecimento da estrutura do mundo no qual a história se desenrola, para poderem então ser resolvidas.

1.3. OBJETIVOS

O objetivo geral deste trabalho foi especificar um sistema de autoria para ambientes educacionais, altamente interativos, compostos de micro-mundos virtuais. Tais micro-mundos são modelados sob o paradigma dos “adventures”, onde o aprendiz é parte integrante deste universo e interage diretamente com este, através do próprio papel que representa, deixando de ser um mero espectador.

O sistema de autoria provê a criação de ambientes virtuais, tridimensionais, multiparticipantes.

Como objetivos específicos, desenvolveu-se um protótipo da ferramenta além de criar também um modelo prototipado de uma possível aplicação na área de engenharia civil. Através desta implementação/protótipo, pode-se avaliar as potencialidades da ferramenta especificada, assim como determinar os requisitos para que um sistema de autoria de micro-mundos seja amigável e, ao mesmo tempo, possibilite a criação de ambientes educacionais que estimulem as funções cognitivas que auxiliam o processo de aprendizado do usuário.

1.4. ESTRUTURA

Este documento está estruturado em onze capítulos. Cada capítulo constitui uma parte essencial do estudo feito para a elaboração deste projeto.

O capítulo 2 apresenta alguns aspectos relevantes sobre o estudo de Jean Piaget, sua teoria cognitiva e também a aplicação desta dentro do trabalho desenvolvido.

O capítulo 3 traz um levantamento de algumas ferramentas de autoria muito utilizadas hoje em dia para facilitar a criação de aplicações educacionais.

O capítulo 4 tem apresenta uma síntese acerca dos ambientes educacionais e dos que se pode chamar de ambientes potencialmente educacionais, sendo considerados aqui os jogos. Neste capítulo, são ainda apresentados, a título de exemplificação, alguns *softwares* educacionais e jogos, alguns dos quais possuem o paradigma de “adventure games”.

Os capítulos 5 e 6 versam sobre a Realidade Virtual e sobre a aplicação desta na Educação.

Os capítulos 7 e 8 trazem a especificação e apresentação da ferramenta de autoria desenvolvida.

Os capítulos 9 e 10 versam sobre a área da aplicação e apresentam o modelo protótipo desenvolvido.

E finalmente no capítulo 11, têm-se as conclusões sobre o trabalho desenvolvido, bem como sugestões para o aprimoramento e extensão do mesmo.

Este documento apresenta ainda um anexo, contendo uma breve descrição das características gerais das linguagens de programação pesquisadas e utilizadas neste projeto.

2. FUNDAMENTAÇÃO TEÓRICA

"O sujeito que conhecemos através da teoria de Piaget é um sujeito que trata ativamente de compreender o mundo que o rodeia e de resolver as interrogações que este mundo lhe coloca. É um sujeito que aprende basicamente através de suas próprias ações sobre os objetos do mundo e que constrói suas próprias categorias de pensamento ao mesmo tempo que organiza seu mundo", [DOL, 87].

2.1. INTRODUÇÃO

O uso da informática na educação deve ser fundamentado em teorias que enfatizem o processo de construção do conhecimento pelo aluno, nesse caso é possível recorrer às teorias cognitivas como a de Jean Piaget.

Como é enfatizado por Ramos, em [RAM 95], " se a perspectiva pedagógica adotada, for opressora, então não haverá aprendizado. Não importa quão maravilhosa seja a ferramenta."

Este capítulo apresenta resumidamente o estudo de Jean Piaget, com o intuito de elucidar a relação da teoria construtivista com a proposta desenvolvida.

2.2. O ESTUDO DE JEAN PIAGET

2.2.1. A Teoria da Equilibração Cognitiva

A principal preocupação de Piaget foi o estudo teórico e experimental da gênese dos processos mentais, ou seja, a compreensão de como esses processos vão sendo construídos ao longo da vida do indivíduo.

As estruturas cognitivas possuem um modo de funcionamento constante durante todo o tempo, por isso são denominadas de **invariantes funcionais**.

As **invariantes funcionais** básicas são duas: a **organização** e a **adaptação**. Biologicamente, as duas são inseparáveis, correspondendo a processos complementares de um mecanismo único, onde a adaptação é o processo externo e a organização o processo interno do ciclo [ULB 94].

Como afirma o próprio Piaget em [PIA 87], “esses dois aspectos do pensamento são indissociáveis: é adaptando-se às coisas que o pensamento se organiza e é organizando-se que estrutura as coisas.”

A **adaptação** ocorre quando um organismo se transforma em função do meio e quando essa variação tem por efeito um acréscimo das trocas em ambos, acréscimo esse favorável à sobrevivência do organismo.

A **adaptação** possui dois componentes: a **assimilação**, na qual o indivíduo atua sobre o meio, transformando-o, a fim de adequá-lo às suas estruturas; e a **acomodação**, na qual o sujeito se modifica para se ajustar às diferenças impostas pelo meio.

Estes mecanismos biológicos de adaptação formam os “**esquemas**”, que são estruturas mentais ou cognitivas, com as quais os indivíduos se adaptam e organizam o meio - ou seja, são padrões de comportamento.

“A vida mental, diz Piaget, consiste num processo de assimilação, com o que ele quer dizer a incorporação de experiência em esquemas pela própria atividade da pessoa e um processo de acomodação pelo qual esses esquemas são continuamente modificados em certa medida, como reação a dados novos. Adaptação ao ambiente envolve a consecução de progressivos estados de equilíbrio entre assimilação e acomodação”, [BRE 73].

Como dito em [WAZ 93], “o processo cognitivo não consiste de uma mera busca assintótica por um equilíbrio inalcançável, justamente porque o ponto de equilíbrio pode se deslocar de acordo com as circunstâncias. Na verdade, ele se parece bem mais com uma série de equilibrações, e desequilíbrios momentâneos seguidos de reequilibrações não menos momentâneas”.

Daí, uma definição de suma importância na teoria de Piaget - a **equilibração majorante**, reequilibrações que buscam um patamar cada vez mais elevado de equilíbrio. Os **patamares de equilíbrio**, são estágios nos quais o equilíbrio cognitivo é parcialmente alcançado. Estes estágios podem se transformar rapidamente em situações de desequilíbrio se houver um deslocamento do ponto ideal para outra região. O que permite afirmar que, a cada

passo, não se pode prever qual será o próximo patamar, pois, na verdade, isso depende de como a interação do sujeito com o meio irá transcorrer.

Sabe-se que a frequência e/ou a força dos desequilíbrios diminui com o decorrer do tempo. Isto ocorre devido às novas estruturas cognitivas que o sujeito constrói o que acarreta na criação de um campo maior de abrangência de seu conhecimento, no sentido quantitativo; e também devido às novas formas de estruturas e novos mecanismos que o próprio sujeito constrói, capazes de prever certas situações antes mesmo que elas aconteçam (esquemas preditivos). Deste modo, o que antes poderia aparecer como uma perturbação, causando desequilíbrio no sistema, pode ser mais tarde previsto, mesmo se vier com outras formas variantes, e assim ser tratado de maneira menos traumática [WAZ 93].

2.2.2. Os Períodos da Construção da Inteligência

Piaget classificou o desenvolvimento cognitivo do indivíduo em quatro principais estágios: sensório-motor, pré-operatório, operatório concreto e operatório formal. Estes estágios se fundem gradualmente uns nos outros, através de numerosos subestádios, e percebe-se que os primeiros estágios são absorvidos pelo pensamento posterior.

As faixas etárias determinadas para cada estágio não representam patamares discretos e também não são encaradas como níveis de amadurecimento ou níveis educacionais, mas como resultado de assimilação e acomodação, que por sua vez dependem da interação entre amadurecimento e experiência [BRE 73].

2.2.2.1. Estágio Sensório-Motor (0 - 2 anos)

Neste estágio a criança aprende a coordenar suas ações com o que percebe ou com outras ações e a usar certos esquemas elementares, isto é, maneiras de comportar-se aplicáveis a muitos objetos diferentes (por exemplo, acompanhar com os olhos, sacudir-se, dar batidas, sugar, etc.).

Durante esse tempo a criança adquire a idéia de permanência de um objeto, mesmo quando ele não está imediatamente à sua frente, e uma compreensão de primeiro nível de

algumas relações em espaço, causalidade e tempo.

O estágio sensório-motor está dividido em seis sub-estágios:

- a) adaptação reflexa (0 à 1 mês);
- b) reações circulares primárias (2 à 4 meses);
- c) reações circulares secundárias (4-5 até 8-9 meses);
- d) coordenação de esquemas secundários (8-9 até 11-12 meses);
- e) elaboração do objeto (11-12 até 18 meses);
- f) início da representação (18 até 24 meses).

2.2.2.2. Estágio Pré-Operatório (2 à 7 anos)

Neste estágio a criança aprende a representar o mundo por meio de sinais e símbolos, isto é, imagens e palavras. Reorganiza constantemente seu retrato do mundo através de brinquedo imaginativo, conversação, indagação, audição e experimentação. Pode ser considerado como uma fase de transição de uma inteligência sem linguagem, representação ou conceitos, para uma inteligência representativa.

“O pensamento da criança pré-operacional é egocêntrico. Predomina uma visão de mundo que parte do próprio eu. Não consegue ‘pensar em seu próprio pensamento’, só vivenciá-lo. Suas explicações e suas crenças baseiam-se na mistura de impressões reais e fantásticas que resultam num entendimento distorcido da realidade”, [FIA 94].

2.2.2.3. Estágio Operatório Concreto (7 à 11 anos)

As crianças são ditas no estágio de operações concretas, quando podem lidar com as propriedades do mundo que imediatamente as envolvem, formando gradualmente os esquemas de conservação de matéria e comprimento, peso e volume mais tarde, e assim por diante. A operação é um meio de organizar fatos já internalizados sobre o mundo real, de modo que eles possam ser usados seletivamente na solução de novos problemas.

Dolle, em [DOL 87], elaborou o seguinte quadro para representar a evolução dos

esquemas de conservação.

7 anos	Conservação das equivalências quantitativas
	Conservação dos comprimentos
	Conservação das superfícies
7-8 anos	Conservação das substâncias
8-9 anos	Conservação do peso
	Primeira conservação do volume espacial, "volume interior"
	Conservação das verticais de horizontais
11-12 anos	Conservação do volume (físico)
	Conservação do volume espacial

Tabela 2.1: Evolução dos esquemas de conservação

2.2.2.4. Estágio Operatório Formal (a partir dos 12 anos)

Durante este estágio as crianças estão aprendendo a atacar problemas do ângulo de todas as combinações possíveis e são capazes de efetuar experimentações controladas, nas quais podem observar o efeito de alterar uma ou mais variáveis de cada vez.

Este estágio envolve operações "formais" assim como concretas, isto é, envolve a capacidade de fazer livre uso de raciocínio hipotético.

"Isto significa que o campo de equilíbrio é infinitamente mais extenso do que nos níveis anteriores e que os instrumentos de coordenação são mais flexíveis. Agora a leitura da experiência já não se efetua apenas por uma apreensão de suas propriedades ou das propriedades das ações de transformação que são alcançadas de uma lógica de classificação e de ordem; ela procede formulando hipóteses que colocam os dados a título de dados, vale dizer, independente de seu caráter atual. Depois ela combina entre si segundo exigências de uma nova lógica ou lógica das proposições, o que equivale a dizer segundo uma lógica de todas as combinações possíveis", [DOL 87].

2.2.3. Comentários acerca da Teoria Construtivista

Sob a perspectiva piagetiana, o pensamento é a base em que se fundamenta a

aprendizagem, e esta é uma construção centrada na pessoa que a realiza. Piaget [PIA 76] revelou que o ser humano tem uma pré-disposição para pensar, julgar, argumentar com bases racionais, e necessita desenvolver estas potencialidades no decorrer da vida. Portanto, é importante ressaltar que o conhecimento é produzido na interação com objetos e seres do ambiente, propiciando o desenvolvimento de esquemas mentais e, por conseguinte, o aprendizado.

A teoria construtivista [NOV 95] propõe que o aluno participe ativamente do próprio aprendizado, mediante a experimentação, a pesquisa em grupo, o estímulo à dúvida e o desenvolvimento do raciocínio, entre outros procedimentos. Esta teoria rejeita a apresentação de conhecimentos prontos ao estudante, como um prato feito. Daí o termo “construtivismo”, pelo qual se procura indicar que uma pessoa aprende melhor quando toma parte de forma direta na construção do conhecimento que adquire.

O construtivismo enfatiza a importância do erro não como um tropeço mas como um trampolim na rota da aprendizagem. O erro passa a ter um caráter “construtivo”, isto é, serve como propulsor para se buscar a conclusão correta.

Para os construtivistas não se aprende por pedacinhos, mas por mergulhos em conjuntos de problemas que envolvem vários conceitos ao mesmo tempo.

2.3. Aprendizagem através da construção do conhecimento

Vários são os fatores que levam a crer que a Teoria Construtivista é adequada aos ambientes de aprendizagem de Realidade Virtual (RV). Através desta tecnologia emergente, consegue-se criar ambientes para a construção do aprendizado. O aprendiz pode ele mesmo vivenciar as experiências passando de um mero espectador à um agente ativo do seu próprio aprendizado. Isto é verificado não apenas em ambientes imersivos mas também naqueles em que o aprendiz atua no papel de personagem do mundo virtual, através de um *avatar*.

Há, é claro, quem defenda que o ponto chave de encontro das duas correntes - uma a teoria pedagógica e outra a tecnológica, é a imersão [WIN 93, CAS 96]. O próprio William Winn em [WIN 93], conclui que o construtivismo provê a melhor teoria para o

desenvolvimento de aplicações de RV, e que a teoria construtivista promove a melhor e provavelmente a única estratégia que permite estudantes a aprenderem de experiências não-simbólicas em primeira pessoa.

Já Christine Byrne em [BYR 96], conduziu um estudo para explorar a RV como uma ferramenta educacional. Trabalhou com estudantes da escola regular que criaram moléculas em um ambiente virtual imersivo. Seus conhecimentos foram testados sobre estrutura atômica e molecular antes e depois de sua experiência com RV. Estes resultados foram comparados aos resultados dos testes de estudantes que experimentaram outro ambiente de aprendizagem sobre o mesmo tópico. O outro ambiente diferenciava-se de RV em termos de imersão e interatividade.

Através deste estudo conseguiu-se concluir que a interatividade foi considerada como significativa, enquanto a imersão foi considerada não significativa. Questões de treinamento, projeto do mundo virtual, resolução de *hardware* e o número de estudantes foram sugeridos como possíveis razões pela falta de significância da imersão neste estudo. Mas, mesmo assim, o fator imersivo, não anulou a relevância da interação.

Na dissertação em questão, assume-se que a imersão não é um fator essencial para ambientes de aprendizagem construídos sob o paradigma construtivista. Não se tem o intuito de menosprezar a potencialidade de ambientes educacionais imersivos, mas sim, o de evidenciar que ambientes educacionais não-imersivos, como o desenvolvido no projeto em questão, também estão intimamente relacionados com a teoria construtivista. Isto pode ser verificado através do paradigma proposto, no qual o aprendiz é um agente ativo do próprio universo do qual faz parte, pois ele “vive” e interage com o micro-mundo todo o tempo, através de atividades relacionadas a este mundo ao qual está inserido. É através destas atividades que o aprendiz fortalece e acelera a construção de suas estruturas cognitivas, ou seja, aprende através desta exploração.

Vem-se a acreditar então que o paradigma construtivista é totalmente aplicável dentro da presente proposta, e através da aplicação destas idéias é que se pretende alcançar um ambiente educacional de excelência.

3. FERRAMENTAS DE AUTORIA

3.1. Introdução

Várias são as ferramentas encontradas para facilitar a criação de aplicações educacionais nas mais diversas áreas de ensino.

Neste capítulo pretende-se comentar em linhas gerais os objetivos e também o suporte que oferecem algumas ferramentas muito utilizadas hoje em dia, dentre as quais pode-se citar: o **Multimedia ToolBook** (Asymetrix), o **IconAuthor** (AimTech), o **HyperCard** (Apple), o **Director** e o **AuthorWare** (ambos da MacroMedia).

Os softwares de autoria podem ser classificados em 3 grupos distintos:

- a) pacotes baseados em telas (páginas ou fichas);
- b) pacotes baseados em ícones ou dirigidos a eventos;
- c) pacotes baseados em linha do tempo e em apresentações.

Nos pacotes baseados em telas o autor trabalha em telas que são WYSIWYG (What You See Is What You Get), ou seja, as telas durante a autoria são reproduções exatas das telas que compõe o aplicativo que está sendo desenvolvido. Nesta categoria se enquadram o **Multimedia Toolbook** e o **HyperCard**. O termo usado para definir uma tela varia. O **Toolbook** denomina cada tela de página, que em conjunto formam a metáfora de um livro. O **HyperCard** usa o termo ficha, numa analogia na qual um programa é uma pilha de fichas.

Os principais pacotes baseados em ícones são o **IconAuthor** e o **AuthorWare**. Nesses softwares, a construção de um aplicativo multimídia é feita através de um fluxograma (diagrama de ícones), e não através de *scripts*. Na janela principal do **AuthorWare** e do **IconAuthor**, o desenvolvedor não vê a tela do seu título, e sim seu fluxograma. Esse esquema representa, de cima para baixo, a seqüência de interações do aplicativo com o usuário. Telas, sons, animações, vídeos e opções de menu são representados por ícones interligados, mostrando os possíveis percursos.

A ação exata de cada ícone é definida através de um duplo clique, que revela o *Content* — — —

Editor (Editor de Conteúdo). É nessa janela que são definidas as imagens a serem exibidas, as áreas sensíveis da tela e a lógica que determinam o fluxo de atividade.

O Macromedia Director representa a categoria de softwares de autoria baseados em linha de tempo. Otimizado para trabalhar com mídias dinâmicas como animação, vídeo e áudio, pode-se dizer que o Director possui a metáfora de uma produção cinematográfica onde o diretor é o próprio autor do aplicativo.

A maior parte do trabalho no Director é feita numa janela chamada *Score* (partitura). As colunas representam seqüências paralelas de ações no mesmo instante de tempo. A unidade de tempo é o *frame* (quadro) que pode representar uma tela estática ou o instante de uma animação. Os objetos colocados no *Score* podem ser vídeos, sons, botões, caixas de texto, etc. Todos os recursos interativos são baseados no *Score*.

Como no Toolbook, também é possível colocar *scripts* em botões, mas na maioria dos casos os comandos são associados a pontos específicos da partitura. Ao lado da janela *Score* o diretor tem acesso ao *Stage* (palco) que é a reprodução da tela apresentada ao leitor. No palco os objetos são posicionados e animados interativamente. Há ainda o *Cast* (elenco), que exhibe todos os objetos incluídos no aplicativo na forma de pequenos ícones. Clicando sobre esses ícones, o diretor pode editar os textos, retocar as imagens, alterar os rótulos, etc.

3.2. Multimídia Toolbook

3.2.1. Histórico

A primeira versão do Toolbook foi criada em maio de 1990 pela Asymetrix Corporation para o Windows 3.0. O Toolbook é descrito pela própria Asymetrix como uma ferramenta de construção de *software* para ser usada como ambiente de desenvolvimento de aplicativos para o Microsoft Windows. Convém lembrar que nessa época o Windows 3.0 não possuía as extensões multimídia, que só foram incorporadas em 1992.

Em 1992 a Asymetrix lançou a versão 1.5 de forma que o Toolbook pudesse utilizar as extensões multimídia do Windows. Poucas alterações foram feitas até a versão 1.53.

Em meados de 1994 foi lançada a versão 3.0 do Toolbook, mantendo a mesma filosofia, ou seja, a metáfora do livro, embora com inúmeros recursos acrescentados e uma performance sensivelmente melhor. Logo a seguir, sai o Multimedia Toolbook 3.0 que oferecia todos os recursos do Toolbook 3.0 mais uma otimização de código para tratar com elementos complexos de multimídia. Novos recursos e objetos (gerenciador de clips, gerenciador de objetos, suporte a animação, edição de vídeo, etc.) vieram trazer funcionalidade e facilidade de uso para multimídia.

No final do ano de 1994, a Asymetrix lança o Multimedia Toolbook 3.0 - CBT Edition, que incorporava características importantes para o desenvolvimento de aplicações de treinamento baseadas em computador. Com o recurso de Course Management System (CMS) os autores são guiados através de livros especialistas, reduzindo o tempo de desenvolvimento e permitindo não só a criação dos mais diversos aplicativos de treinamento (tutorial interativo, simulação de provas, treinamento empresarial, etc.) como também acompanhamento do desempenho dos alunos.

Em 1996, foi lançada a última versão do Multimedia Toolbook - a versão 4.0, que fornece diversas ferramentas necessárias à criação de aplicações gráficas que combinam interatividade, manipulação de textos e dados, em conjunto com gráficos, animação, áudio e vídeo.

A seção seguinte, levanta algumas características gerais acerca do Multimídia Toolbook e apresenta alguns aspectos inovadores da nova versão.

3.2.2. Características Gerais

O Multimedia Toolbook é considerado um ambiente de desenvolvimento orientado a objetos que possui ferramentas sofisticadas e uma linguagem de programação clara e poderosa - *OpenScript*.

O Toolbook usa a metáfora de um livro para definir um aplicativo. Uma aplicação do Toolbook consiste de arquivos que se chamam livros. Um livro consiste de páginas e *background*. Páginas contém campos-texto, imagens, botões, gráficos entre outros itens que

são os objetos. Cada página pode ter diferentes objetos responsáveis pela ligação (*link*) entre elas. Um *link* pode ser criado a partir de um botão ou então através de um *hotword*. *Hotword* é um campo ou palavra destacada no texto, que, após ser clicada no modo leitor, esta permite a visualização de uma nova página.

No Toolbook existem dois níveis de trabalho: modo leitor e modo autor. No modo autor, o usuário através das ferramentas e objetos disponíveis, pode dar vazão às suas idéias e criar seus projetos. Já no modo leitor, o usuário passa a ser um mero explorador, e não mais um criador, percorrendo sistemas já implementados ele explora suas possibilidades. A grosso modo, no modo autor, constrói-se um livro eletrônico e no modo leitor, lê-se o livro.

O poder da linguagem de *scripts* do Toolbook, aliado à possibilidade de *linkar* bibliotecas DLLs desenvolvidas a outros aplicativos permite a construção de apresentações com alta qualidade e interatividade. Além de poder usar bibliotecas DLLs para executar funções especiais em outras linguagens para Windows, o Toolbook permite que se utilize DDE para enviar dados ou mensagens para outros programas Windows. Possui também a possibilidade de escrita em arquivos e ainda a possibilidade de poder chamar arquivos .com, .exe, .bat ou arquivos no formato .win.

Entre os tipos de aplicações desenvolvidas no Toolbook pode-se citar:

- a) Quiosques de informações interativos;
- b) Treinamento e tutoriais interativos;
- c) Documentos hipermídia, hipertexto;
- d) Aplicações *front-end* para bancos de dados;
- e) Demonstrações de produtos on-line;
- f) Jogos e aplicações educacionais.

3.2.3. Multimídia Toolbook 4.0 (MTB 4.0)

O Toolbook possui seu próprio editor de textos (podendo importar ASCII e arquivos .rtf); seu próprio editor gráfico (podendo importar arquivos .dib, .cgm, .drw, .eps, .tif, .gif Windows Metafile, e Windows bitmap; pode importar também .jpg se usado com Quick Time

for Windows); editor de som (requer placa de som executando MIDI, WAV e VOC); editor de animação, editor de estrutura (salto por resposta individual, busca por palavra chave) e outros.

O MTB 4.0 possui várias características extras em relação às versões anteriores, onde pode-se citar:

- a) Compatibilidade com Windows 95, Windows 3.1 e Windows NT: os objetos de interface e controles do Windows 95 são suportados no Multimedia ToolBook. Os controles Windows são apresentados de forma diferente, dependendo de qual dos ambientes Windows a aplicação é executada (Windows 3.1, 95 ou NT);
- b) Interoperabilidade com aplicações Windows: a importação de controles Visual Basic® (VBX) estende o uso de objetos Windows: campos de dados personalizados, grades, relógios, timers e outros. Controles VBX são pré-programados habilitando a incorporação de capacidades sofisticadas personalizando uma aplicação com pouca ou nenhuma programação;
- c) MTB 4.0 apresenta *runtime* 50% mais rápido que a versão 3.0;
- d) *Scripts* Compartilhados: podem ser usados para associar o mesmo comportamento à múltiplos objetos. *Scripts* compartilhados economizam tempo e simplificam a manutenção de aplicações;
- e) Editor de propriedades - *Property Editor*: permite a rápida visualização e edição de propriedades dos objetos;
- f) Otimizador automático de paleta - *Palette Optimizer*: o otimizador de paleta automaticamente cria uma paleta comum com as cores dominantes de todos os gráficos e vídeo de um aplicativo, garantindo transições suaves entre páginas;
- g) Janela de propriedades dos objetos - *Object Browser*: os objetos são mostrados hierarquicamente permitindo seleção, edição e deleção;
- h) Controles integrados com URL's da Internet.

4. SOFTWARE EDUCACIONAL

4.1. INTRODUÇÃO

Este capítulo discute duas linhas gerais acerca do conceito de *software* educacional, ou seja, aplicações com fins educativos evidentes, e o que se pode chamar de ambientes potencialmente educacionais, sendo considerados aqui os jogos.

4.2. JOGOS

Os jogos de uso geral que buscam diversão e entretenimento através de brincadeiras, quebra-cabeças, ou até mesmo combate, podem ser tratados como ambientes potencialmente educacionais por não possuírem *a priori* um fim educativo. Mas existe por trás deste tipo de ambiente um ganho educacional significativo, no que concerne ao estímulo de diversas habilidades por parte do jogador.)

Os jogos promovem o desenvolvimento de habilidades de pensamento e raciocínio, e incorporam motivação ao ambiente, oferecendo assim de uma maneira natural, meios para desenvolver tais habilidades. Com jogos aprende-se também a cooperar, barganhar, respeitar e projetar conseqüências de longo prazo em um cenário.

Jogos educativos são considerados potencializadores de motivação no reforço de habilidades e informações previamente ensinadas [LER 94]. Um único jogo pode desenvolver e aperfeiçoar diversos tipos de conceitos, sendo aplicável em diversas áreas do conhecimento.

4.2.1. Jogos via Internet

Os jogos como Galactic Bloodshed, Empire e Multi-User Dungeons (MUDs), são jogos de representação (RPGs - *Role Playing Games*) ou simulação de aventuras. Os aficionados costumam chamá-los de “aventuras de realidade virtual baseadas em texto” [LAQ93].

Os jogos podem simular batalhas, armadilhas para pegar incautos e até mesmo magia. Os jogadores interagem em tempo real e podem alterar o “mundo” do jogo de que participam.

Os jogos exigem que o usuário se dedique a um intenso aprendizado de todos os personagens e peculiaridades do jogo, isso sem mencionar as regras. Na maioria dos jogos, os jogadores iniciantes assumem um personagem e passam a atuar no jogo.

A próxima sessão apresenta uma breve explicação sobre os MUDs.

4.2.1.1. MUDs

O termo MUD [FAQ 96b] é empregado para conceituar '*Multi-User Dimension* ou *Multi-User Dungeon* ou até mesmo *Multi-User Dialogue*'. MUDs são programas do tipo jogo, sob o paradigma de RPGs - "*Role Playing Games*", multi-usuários, jogados com outras pessoas ao redor do mundo via acesso remoto.

O ambiente de um MUD é um micro-mundo virtual, particionado em várias salas virtuais - "*rooms*", onde cada usuário tem o controle de um personagem, e através deste, pode caminhar, conversar com outros personagens (outros usuários), explorar áreas, manipular objetos, resolver quebra-cabeças, e até criar sua própria "*room*", descrições e itens.

Os personagens e objetos não podem interagir diretamente com personagens e objetos que estejam em outra sala, a interação é feita através de texto, e a comunicação entre eles geralmente é manipulada por *sockets TCP*.

Os personagens no MUD incluem desde heróis de ficção científica, animais peludos, monstros, bruxos, até cavaleiros da época medieval, tudo com muito mistério.

Os objetos podem ter muitas formas, por exemplo armaduras, armas, bastão mágico, recipiente para bebidas, chaves, e até mesmo comida. Alguns poderão ser usados, alguns levados consigo, alguns deixados para trás e alguns empunhados. Os objetos são manipulados através de diversos comandos.

Na Tabela a seguir estão listados a sintaxe e descrição de alguns comandos genéricos relacionados aos objetos.

SINTAXE	DESCRIÇÃO
get[object name]	permite pegar objetos para serem utilizados depois
put[object name][container]	permite colocar objetos em 'containers' no qual pode-se levar consigo
look[object name]	permite ver características especiais de um objeto
open[object name]	este comando é utilizado para os objetos que estão nos 'containers', onde alguns poderão ser retirados e destes um poderá ser aberto
close[object name]	permite fechar um objeto que esteja aberto

Tabela 4.1: Lista de comandos relacionados aos objetos

Nos MUDs em geral, além dos comandos relacionados ao objeto, têm-se também comandos relacionados à *room* e comandos de comunicação com outros usuários. Para exemplificar a Tabela 4.2 apresenta uma pequena lista de comandos relacionados à *room*.

SINTAXE	DESCRIÇÃO
look[room door etc]	permite visualizar a <i>room</i> e o que ela contém
open[door name][direction]	permite abrir a porta para outra sala
close[door name][direction]	permite fechar a porta de uma sala
unlock[door name][direction]	permite destrancar uma porta caso possuir a chave correta
pick[door name]	permite destrancar algumas portas

Tabela 4.2: Lista de comandos relacionados à *room*

4.2.1.2. Clientes

Um programa cliente é simplesmente um programa utilizado para fazer a conexão com um MUD. Estes programas geralmente são escritos em linguagem C e estão associados aos servidores. Telnet é um tipo de programa cliente.

Os clientes também provêm vários utilitários, até mesmo macros e habilidades para

impedir ou liberar certas saídas no MUD. Os clientes geralmente são executados pelo UNIX ou pelas versões BSD e SysV, alguns são executados sobre VMS com MultiNet ou Wollongong Networking, e há também, clientes para Macintosh e PC executando WinSock.

Existe uma lista razoavelmente precisa dos diferentes clientes disponíveis hoje em dia. Esta lista é apresentada a seguir.

- a) Clientes UNIX: TinyTalk, Tiny Fugue, TciTT, VT, LPTalk, SayWat, PMF, TinTin, TinTin++, Tush, Lpmudr, Muddle.
- b) Clientes Emacs: MUD.el, TinyTalk.el, Lpmud.el, CLPmud.el, MyMud.el.
- c) Clientes VMS: tfVMS, TINT, TINTw, DINK, FooTalk.
- d) Clientes WinSock: VWMud, WinWorld, MUTT, MudWin, MUDSock, Pueblo, zMUD, AvPlay, GMUD, VTW, MUSHClient, Phoca.
- e) Clientes Macintosh: MUDDwelle, Mudling, MUDCaller.
- f) Outros Clientes: REXXTALK, RXLPTalk, NTTinTin, WinTin, Clientes BSXMUD.

4.2.1.3. Servidores

No mundo MUD, o servidor detém o caminho da base de dados, os jogadores correntes, as regras e algumas vezes o tempo ou o “*heartbeat*”.

Na maioria das vezes o UNIX é o sistema operacional que executa o servidor, podendo também ser executado por BSD ou SysV.

Existem diversos tipos de servidores, acredita-se até na possibilidade de ter tantos servidores quanto os diferentes MUDs. Na verdade, há essencialmente três principais grupos. A seguir é apresentada uma lista contendo os principais MUDs de cada grupo.

- a) MUDs orientados ao combate - “*Combat-oriented MUDs*”: MUD, AberMUD, DGD, LPMUD, DikuMUD, YAMA, UriMUD, Ogham, CircleMUD e AmigaMUD.
- b) MUDs orientados a recreação - “*Social-oriented MUDs*”: TinyMUD, TinyMUCK, TinyMUSH, PennMUSH, AlloyMUSH, TinyMUSE, TinyMAGE, MUG,

TeenyMUD e TinyMUX.

- c) MUDs mistos - "*Miscellaneous MUDs*": UberMUD, MOO, LambdaMOO, SMUG, UnterMUD e Mordor.

O termo "*combat-oriented*" geralmente significa que o combate é uma parte essencial da cultura do MUD. Neste caso, o personagem conta com diferentes tipos de armas, armaduras, poções mágicas, etc.; e geralmente deve lutar contra bandidos, monstros ou até mesmo bruxos.

Um MUD orientado à recreação/divertimento possui um foco diferente, ele até pode incluir algum combate, mas o papel a ser desempenhado pelo personagem geralmente é fazer qualquer atividade lúdica ou simplesmente conversar com amigos. Os jogadores deste tipo de MUD tendem a se agrupar, conversar, encontrar amigos, fazer brincadeiras e discutir sobre todo tipo de coisas.

A maioria dos MUDs mistos não são MUDs orientados ao combate. Apesar de que muitos se basearam em MUDs orientados ao combate em muitas coisas, esta categoria não é uma derivação direta daquele tipo.

4.2.2. Jogos Monousuários

4.2.2.1. TIM - *The Incredible Machine*

TIM é um jogo desenvolvido pela empresa Sierra, monousuário, para ambiente DOS, que possui alto poder educativo embutido, desenvolvendo o raciocínio em atividades relacionadas a resolver quebra-cabeças utilizando para isto conhecimento sobre leis da física em geral (como mecânica, óptica, cinética, etc.).

Para cada quebra-cabeça, o programa oferece um conjunto limitado de ferramentas para sua resolução. O aspecto lúdico do jogo está propriamente contido no tipo de ferramenta que é oferecida, que pode ser observado na Figura 4.1.

Possui também um modo de edição que permite ao usuário criar novos quebra-cabeças e incluí-lo ao conjunto dos já existentes.

O programa possui várias fases, em que o usuário deve resolver o problema ali apresentado para mudar de fase, possui uma contagem de tempo, que não é essencial, porém um tempo maior gasto na resolução do problema implica numa menor pontuação.

Pode-se dizer que a tela inicial de uma fase é constituída por quatro áreas principais:

- a) Painel de controle - possui ferramentas para controle de som, reinício do jogo, e para modo editor; possui também a gravidade e a pressão atmosférica utilizada neste quebra-cabeça;
- b) Área de trabalho (propriamente dita) - onde é construído o quebra-cabeça;
- c) Área de informação - nesta área é descrito o objetivo do quebra-cabeça;
- d) Área de ferramentas - disponibiliza as ferramentas, juntamente com uma quantidade limitada e suficiente a serem utilizadas para resolver o quebra-cabeça.

Ao ser ativado o início do jogo, entra-se numa outra página que contém apenas a área de trabalho, a área de ferramentas e uma terceira área que exibe a contagem de pontos.

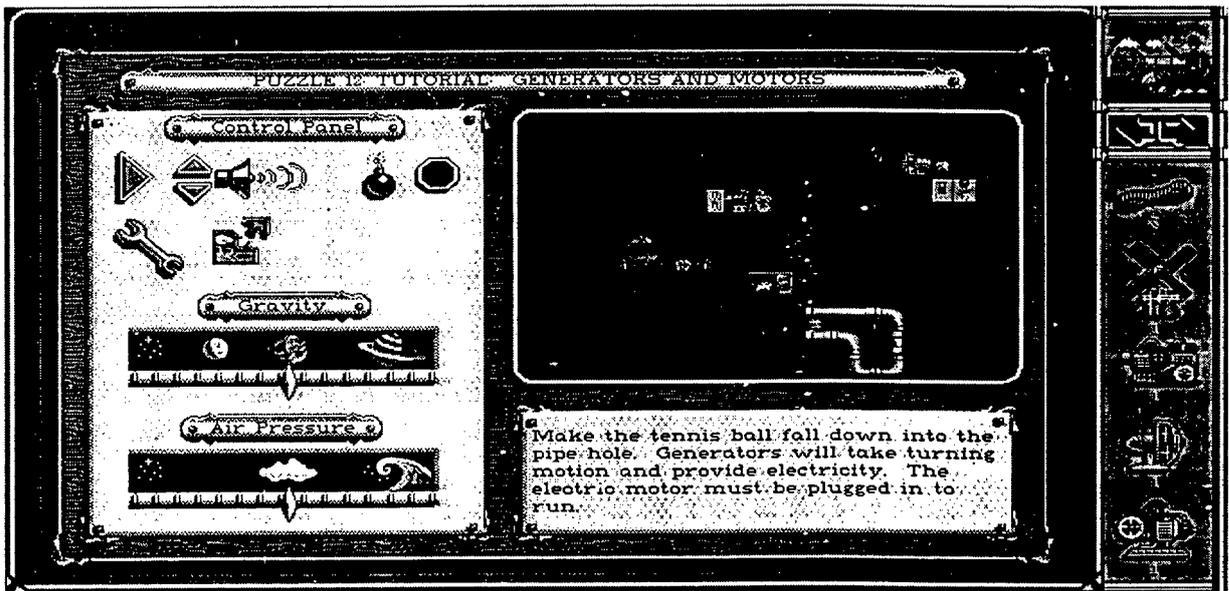


Figura 4.1: Tela do jogo TIM

4.2.2.2. SIMCITY

Simcity é um jogo para ambiente DOS, Windows (3.1 e 95), OS-2 e Macintosh, que

visa desenvolver a capacidade administrativa do usuário. O jogador inicia o jogo com uma quantidade inicial de dinheiro e deverá investir seus recursos na urbanização de uma área. Consiste principalmente em criar uma estrutura física (ruas, água, zoneamento, eletricidade) e uma infra-estrutura básica (escolas, hospitais, postos policiais, corpo de bombeiros, áreas de lazer, espaços culturais e transportes em geral) para áreas habitacionais, comerciais e industriais.

A interação ocorre de diversas maneiras: através de manchetes de jornal; de gráficos que revelam a taxa de descontentamento da população em relação à administração; de gráficos parciais sobre a estrutura física e de demonstrativos de índices de criminalidade, poluição e outros.

O usuário pode controlar o jogo (crescimento da cidade) pela manipulação de taxas de impostos e investimentos nas diversas áreas básicas. Havendo uma boa administração a cidade tende a expandir-se, caso contrário, pode ocorrer um esvaziamento nas áreas já ocupadas, causando uma queda de arrecadação, a qual criará grandes dificuldades para o jogador.

Com o passar dos anos, novas alternativas de energia e transporte, entre outros itens, são oferecidas ao jogador, pois o jogo é iniciado no ano de 1900, quando existiam poucos recursos.

De um modo geral pode-se afirmar que as formas de exploração e interação deste jogo são infinitas, onde o jogador se encontra totalmente imerso na problemática em questão. Além do Simcity, pode-se encontrar o SimFarm (administrar uma fazenda) e o SimEarth (governar o mundo), entre outros que são do mesmo padrão.

4.3. AMBIENTES EDUCACIONAIS

Diversos são os ambientes com fins educativos existentes hoje em dia, portanto, este capítulo não tem a intenção de descrever todos estes - o que seria uma tarefa um tanto tediosa e quase impossível. Cabe aqui apenas descrever alguns softwares educacionais que possuem a mesma abordagem do projeto em questão, que é a de ambientes do tipo “adventures”, ou também chamados de “role playing games”.

4.3.1. O Projeto EPES

O projeto EPES - *European Pool of Educational Software*, é um projeto que une vários países europeus com o objetivo de suprir a necessidade de *software* educativo nas escolas, através do intercâmbio destes.

Os programas produzidos se enquadram em várias séries temáticas das quais se destacam: Ciências da Terra e da Vida, Língua e Idiomas, Tecnologias, Ciências Sociais, Matemática e Ensino Especial.

Através da análise de vinte e cinco softwares educacionais, que constituem parte integrante do projeto EPES do ano de 1995, pôde-se fazer um levantamento e um sucinto relatório sobre as principais características de cada programa.

Serão apresentados a seguir dois programas que possuem um paradigma relacionado à proposta do trabalho em questão.

4.3.1.1. *Software 'ALICE'*

O programa 'Alice' é um ambiente educacional do tipo "adventure", baseado na obra literária 'Alice no país das maravilhas'. Este ambiente faz parte da série temática Línguas e Idiomas do projeto EPES [HOL 93], buscando o aprendizado da língua inglesa. Um dos objetivos para a construção deste *software* é estimular a leitura da obra literária.

Neste ambiente, o usuário é o próprio personagem principal da história - ou seja, a Alice. Este se envolve em diversas atividades (não totalmente iguais à da obra original, pois senão já se saberia como proceder), sem necessidade da mediação constante por parte do professor/orientador, que passa a ter um papel de esclarecedor de eventuais dúvidas que lhe possam surgir.

No papel de Alice o usuário passa por diversas peripécias, só conseguindo sair de algumas situações se levar consigo e usar corretamente os objetos de que dispõe. Terá ainda que se comunicar com as criaturas que encontrar. Ou seja, nesta aventura em cada ambiente que o usuário se encontrar haverá objetos disponíveis e também frases prontas (em inglês)

para este se comunicar com os outros personagens que estão no mesmo ambiente. O usuário deve saber como se comportar em relação a cada personagem e também aos objetos, construindo assim, a partir de suas tentativas e erros, seu próprio conhecimento.

O programa conta com um menu de opções que pode ser visualizado na Figura 4.2, as quais possuem como função:

- a) Rosa-dos-ventos - permite escolher a direção que pretende-se tomar;
- b) Take (pegar) - permite selecionar um dos objetos disponíveis, o objeto selecionado é colocado no cesto à direita do monitor, podendo também, ao invés disto, surgir uma mensagem avisando que este objeto não pode ser selecionado;
- c) Drop (abandonar) - permite libertar-se dos objetos que estão no cesto;
- d) Give (entregar) - esta opção é utilizada quando se deseja entregar alguns objetos que a Alice transporta consigo, para algum outro personagem;
- e) Use (utilizar) - como o próprio nome diz, esta opção permite utilizar um objeto que está em posse da Alice;
- f) Look (examinar) - permite examinar com mais detalhe o local onde se encontra;
- g) Speak (falar) - utilizada para se comunicar com os outros personagens que encontra no decorrer da história;
- h) Help (ajuda) - com esta opção, pode-se obter sugestões sobre o que fazer na situação em que se encontrar;
- i) Save (salvar) - permite salvar as cenas percorridas e carregá-las posteriormente quando desejar continuar o jogo;
- j) End (fim) - finalizar o jogo.

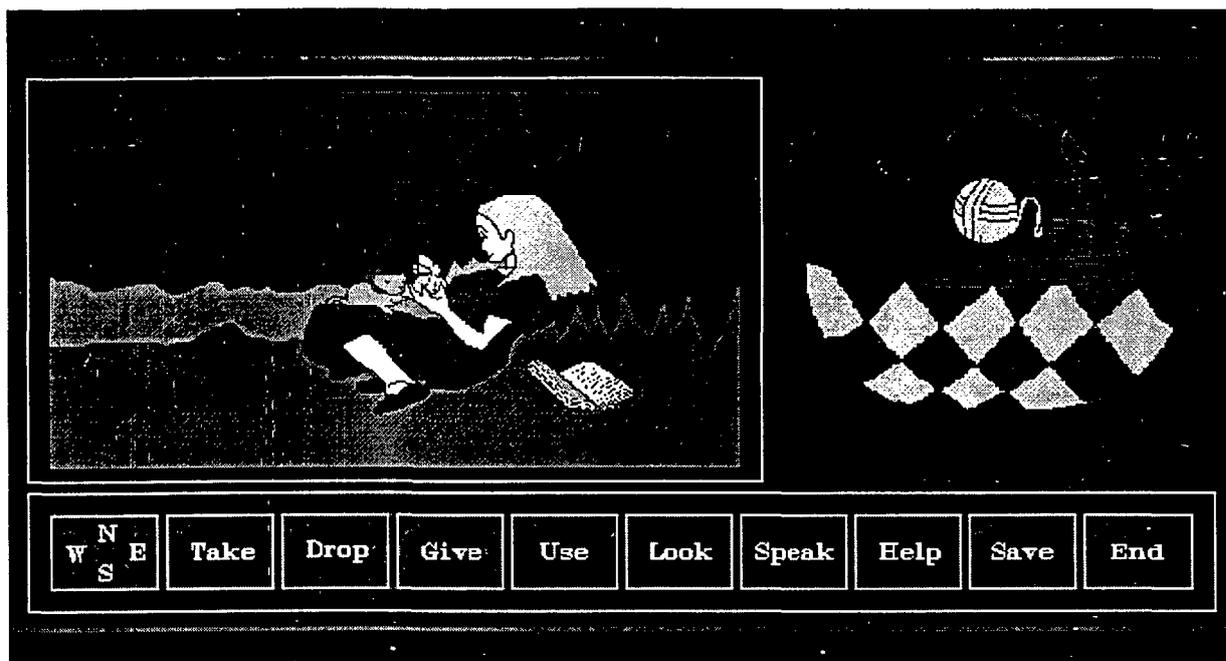


Figura 4.2: Tela inicial do *software* Alice

4.3.1.2. *Software* 'A ALDEIA VIKING'

'A ALDEIA VIKING' é um sistema do tipo hipertexto, mas muito interativo, onde o usuário se torna parte integrante da história - no caso, o próprio chefe da aldeia Viking. Este sistema é parte constituinte da série temática Ciências Sociais do projeto EPES [GAL 93], buscando o aprendizado da história da época dos Vikings.

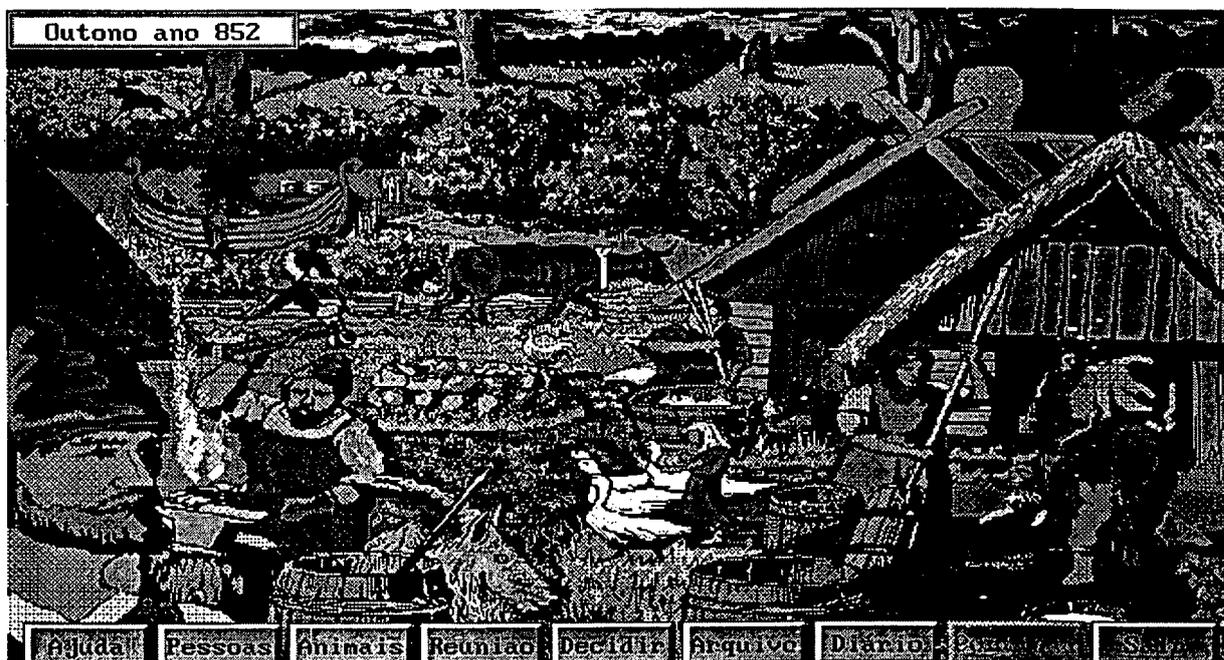
O aprendiz deve já possuir um conhecimento relevante acerca dos Vikings para poder tomar decisões corretas como chefe da aldeia. Isto não implica necessariamente em dizer que o programa não traz nenhum aprendizado, ao contrário disto, o programa põe em prática o conhecimento já adquirido, fazendo o aprendiz refletir, discutir (quando estiver jogando em parceria) e analisar profundamente o problema que ele tem em suas mãos. Através dos enganos cometidos pelo aprendiz e conseqüentemente pela sua própria avaliação, este tem a oportunidade de (re)construir seu conhecimento.

A idéia básica do programa é que o aprendiz se envolva tanto intelectual como emocionalmente no papel de chefe. Nesse sentido é muito importante que ele registre tudo o que acontece no diário. O diário tem como função principal a de auxiliar o usuário para uma

avaliação futura e também auxiliá-lo quando este retornar numa jogada já iniciada anteriormente.

A interação ocorre através de tomadas de decisão que o chefe (usuário) precisa realizar juntamente ao conselho da aldeia, que por sua vez pode votar uma decisão do chefe, e até mesmo destituí-lo do posto, caso este não saiba administrar a aldeia corretamente.

O sistema possui uma base de dados referentes à quantidade de animais existentes por espécies; a distribuição da população em homens, mulheres, crianças, idosos; etc.; além de permitir ao usuário através de sua exploração pelo sistema, resgatar várias informações sobre o modo de vida dos habitantes, tipo de armas usadas, alimentação, vestuário, entre outras, que o auxiliarão na tomada de decisão. A maioria dos objetos da tela são *links* para informações mais detalhadas.



- Figura 4.3: Apresentação de uma das telas iniciais do *software* Aldeia Viking
- Descrevendo rapidamente a função de cada botão da tela da Figura 4.3, tem-se:
- Ajuda - possui informações para auxiliar o usuário (corresponde à tecla F1);
 - Pessoas - determina a distribuição da população, dividida em idosos, homens, mulheres, crianças e bebês recém-nascidos;

- c) Animais - apresenta a distribuição do gado da aldeia, entre vacas, ovelhas/cabras, frangos/gansos e porcos;
- d) Reunião - permite ao usuário encontrar-se com os diferentes conselheiros para ser aconselhado sobre as decisões a tomar acerca da divisão de trabalho, a caça, a pesca, a criação e o abate de animais, entre outros; podendo também ser interrogado pelos conselheiros;
- e) Decidir - baseado na opinião dos conselheiros poderá tomar as decisões;
- f) Arquivo - esta opção serve tanto para salvar a jogada para um futuro retorno, como também para continuar de uma sessão anterior;
- g) Diário - permite escrever sobre os acontecimentos na aldeia, decisões tomadas, resultados, etc.

4.3.2. Software Rocky's Boots e Software Robot Odyssey

A empresa *The Learning Company* é a responsável pela concepção do *software Rocky's Boots* [ROB 84] e de sua continuação *Robot Odyssey* [WAL 84], ambos *adventures* educativos.

Rocky's Boots consiste de um tutorial sobre circuitos combinacionais e seqüenciais no qual o aprendiz é colocado em um ambiente de manipulação direta de peças (portas lógicas, etc.) que podem ser combinadas para formar máquinas simples. No final do tutorial são propostos vários jogos lógicos, onde, para vencer, o aprendiz deve compreender o funcionamento das portas lógicas e saber construir circuitos adequados para a resolução de cada problema específico.

O *software Robot Odyssey* coloca o aprendiz em um labirinto (a cidade dos robôs) dentro do qual seres humanos não são admitidos. O aprendiz tem à sua disposição três robôs nos quais ele pode entrar e usar como veículo para percorrer o labirinto. Os robôs são providos de sensores e atuadores e o aprendiz tem à sua disposição uma caixa de ferramentas contendo um soldador e vários tipos de circuitos lógicos. O aprendiz pode, inclusive, projetar e implementar micro-chips para utilizar em seus robôs. Em cada sala do labirinto diferentes problemas têm que ser resolvidos para que se possa passar. Estes problemas só podem ser

resolvidos quando o aprendiz interliga os circuitos de seu robô de forma correta. Convém salientar que não existe apenas uma forma correta. Qualquer circuito que funcionasse no mundo real também funcionaria no ambiente. Desta forma o aprendiz tem muitos graus de liberdade para criar em cima dos conteúdos aprendidos. A aprendizagem e prática ocorrem simultaneamente neste ambiente.

5. REALIDADE VIRTUAL

"A realidade virtual representa a evolução da tecnologia de interface com computadores, proporcionando um mecanismo de interação que cria um meio mais interessante de trabalhar, aprender e se entreter com computadores. Possui aplicação em diversas áreas, com destaque em treinamento e educação. Nestes casos, requer-se a utilização de agentes inteligentes que atuam como participantes "humanos" auxiliando e monitorando as atividades do usuário, assistindo-o quando necessário, principalmente em ambientes complexos."
[RUS 95]

A Realidade Virtual (RV) tem sido definida em [PAN 93], como um ambiente multimídia baseado em computador, altamente interativo, no qual o usuário torna-se um participante através do computador de um mundo "virtualmente real". Num ambiente virtual, o usuário não parece estar distante da tela do computador, mas torna-se parte da ação na tela, dando a sensação de participação.

RV é definida também como uma simulação gerada por computador de um mundo real ou imaginário. Ela pode ser gráfica (quando possui a representação gráfica do mundo) ou textual (quando possui a descrição textual do mundo).

A realidade virtual pode ser considerada como uma ferramenta para visualizar, manipular, explorar, interagir e modificar - através do computador, dados extremamente complexos de uma forma natural, muito semelhante ao que se faria no caso da ação sobre o dado real. Vista deste ângulo, pode-se salientar quatro características principais dos sistemas de realidade virtual:

- a) imersão;
- b) interação;
- c) envolvimento;
- d) grau de realismo.

A **imersão** impõe ao usuário a sensação de estar dentro do ambiente. Seus movimentos refletem-se dentro do ambiente através de diversos dispositivos, tais como o capacete, a luva, entre outros.

A **interação** é definida como a capacidade que o sistema de RV tem, de responder ao

usuário. Esta interação depende de três fatores:

- a) capacidade do computador para detectar as ações do usuário;
- b) capacidade de modificar o mundo virtual com rapidez;
- c) capacidade reativa do sistema, ou seja, de como o sistema consegue gerar ações sobre os sentidos do usuário.

Neste último item cabe verificar, na análise de um ambiente de RV, quais são os dispositivos que o usuário pode manipular e qual a gama de ações estes conseguem captar ou gerar sobre este usuário.

Já o **envolvimento**, está ligado à capacidade que o sistema de RV possui de cativar ou engajar o usuário para uma certa atividade. De todos os aspectos envolvidos este é o que está menos vinculado ao lado tecnológico, depende isto sim, da criatividade de quem projeta o ambiente. Um exemplo disto são os programas de multimídia, que muitas vezes, mesmo com toda parafernália técnica disponível, não conseguem envolver o usuário por muito tempo.

E, por último, o **grau de realismo**, este se refere à qualidade da imagem, do som, do tempo de resposta, etc., ou seja, neste item avaliam-se todos os aspectos do ambiente pertinentes a quão real o sistema se apresenta.

Várias são as definições de RV encontradas na literatura, muitas delas são criadas em função do trabalho que se está desenvolvendo. No projeto em questão, a RV é definida como um ambiente 3D, multiparticipante e altamente interativo. O usuário é representado fisicamente no ambiente através de um *avatar*, dispensando a utilização de dispositivos para imersão. Este tipo é chamado de *RV Desktop*, o que significa dizer que, o ambiente 3D é visualizado através de um monitor 2D.

A expressão *avatar*, amplamente utilizada para representar personagens em ambientes 3D, significa na meditação religiosa Indiana uma manifestação de um deus ou um ser espiritual na terra.

Nos sistemas de RV a representação humana é um fator altamente relevante. A aparência estática não é muito convincente, os movimentos, as expressões e, mais importante

ainda, as características humanas são cruciais.

Pode se dizer que o uso de *avatar* é o próximo passo para representação pessoal na Internet. Combinado com os objetos eles representam todo o aspecto do mundo “real”.

5.1. Classes de Sistemas de Realidade Virtual

Conforme Pinho, em [PIN 96a], os sistemas de realidade virtual são classificados em cinco categorias:

- a) *Window on World Systems*;
- b) *Video Mapping Systems*;
- c) Sistemas Imersivos;
- d) Sistemas de Telepresença ou Teleoperação;
- e) Sistemas de Realidade Aumentada.

Os *Window on World Systems* ou **Sistemas com Janelas para o Mundo**, são aqueles nos quais o usuário vê o universo virtual por uma tela convencional de computador. Nesta classe enquadram-se hoje a grande maioria dos jogos eletrônicos, os sistemas interativos de navegação além de softwares educacionais. O som destes ambientes é produzido por caixas de som como as dos tradicionais *kits* multimídia. Os sistemas desenvolvidos dentro desta classe não utilizam equipamentos para imersão.

Os **Sistemas de Video Mapping** concentram-se na interpretação dos movimentos do usuário [KRU 91] sem que este precise usar dispositivos ligados ao corpo. A idéia nestes sistemas é capturar, através de uma câmera de vídeo, os movimentos do usuário e a seguir interpretá-los, usando processamento de imagens e reconhecimento de padrões [FOL 92].

Nos **Sistemas Imersivos** o usuário se sente dentro do ambiente virtual. Para tanto, veste um capacete de realidade virtual, luva, rastreador de posição e fones de ouvido. Com estes equipamentos o usuário é “desligado do mundo real” e passa a visualizar, ouvir e sentir apenas os estímulos gerados pelo computador. As ações do seu corpo são interpretadas como a única forma de entrada dos dados [ISD 96].

Os **Sistemas de Telepresença** ou também chamados de **Teleoperação**, são aqueles em que o usuário é colocado em um ambiente real sem de fato estar presente nele. A idéia é que as ações do usuário alterem um mundo físico real diferente daquele onde ele se encontra. Estes sistemas são muito utilizados para treinamento ou manipulação de objetos à distância.

Os **Sistemas de Realidade Aumentada** fazem uma espécie de “fusão” da imagem de um ambiente real com uma informação gerada por computador. Com sistemas deste tipo é possível, por exemplo, olhar para uma impressora e “ver o seu interior”. A idéia é que o sistema de RV capture a imagem da impressora (com uma câmera colocada na cabeça do usuário), faça a fusão desta imagem com a imagem gerada a partir do modelo da impressora (criado em um programa de CAD) e por fim coloque esta imagem, resultante da fusão, no visor de um óculos que o usuário esteja usando.

Um outro uso deste tipo de sistema é a consulta de normas técnicas para manutenção. A empresa *Boeing* por exemplo, está testando um sistema [HED 96] onde os manuais de seus equipamentos aparecem em um visor acoplado ao óculos do técnico que está realizando o conserto, eliminando assim, a necessidade de que ele tire os olhos do equipamento quando precisar fazer uma consulta.

5.2. Dispositivos de Realidade Virtual

Atualmente existem diversos tipos de dispositivos para sistemas de RV. Nesta seção, pretende-se apenas apresentar alguns destes, salientando suas principais funções e características.

A função básica dos dispositivos de RV é produzir no usuário a sensação de imersão em um ambiente virtual. Para isto, eles atuam de duas formas:

- a) lendo os movimentos realizados pelo usuário (ou seja, pelas várias partes do seu corpo), o qual pode ocorrer através de: (i) leitura da posição de um ponto no corpo do usuário (rastreamento), ou (ii) leitura do ângulo de flexão ou rotação de um membro ou parte do corpo do usuário;
- b) impressionando seus sentidos a fim de simular sensações, o qual ocorre em geral

sobre a visão, a audição e o tato.

5.2.1. Dispositivos de Visão

5.2.1.1. Stereo Glasses ou Shutter Glasses

Estes óculos são utilizados em aplicações como visualização científica ou cirurgias nas quais várias pessoas precisam observar a mesma imagem estereóscópica. Estes dispositivos buscam gerar estas imagens a partir de uma tela de computador padrão.

A idéia básica é colocar nos usuários pares de óculos com lentes de cristal líquido capazes de bloquear sua visão quando necessário. O sistema funciona do seguinte modo:

- a) exibe-se na tela a imagem correspondente à do olho esquerdo e bloqueia-se a visão do olho direito;
- b) a seguir faz-se o contrário, ou seja, exibe-se a imagem do olho direito e bloqueia-se a visão do esquerdo.

A maior dificuldade neste sistema, é garantir o sincronismo no processo de exibição, a fim de que o usuário não perceba o funcionamento de tal processo.

5.2.1.2. Head Mounted Displays

Os *Head Mounted Displays* (HMD) ou também conhecidos como VPC - Visores Presos à Cabeça, exibem as imagens de uma cena virtual em duas pequenas telas (uma para cada olho). Estes dispositivos podem ser construídos com dois tipos de monitores, os CRTs ou monitores de TV e os monitores de cristal líquido (LCDs). Enquanto os CRTs são pesados e exibem imagens de alta resolução com altas voltagens muito próximas à cabeça do usuário, os LCDs, por sua vez, são leves, podem ser usados com pequenas voltagens, mas com resolução baixa.

Geralmente, existem sistemas de rastreamento da posição da cabeça acoplados aos HMDs, a fim de permitir a atualização das imagens do mundo virtual de acordo com a direção para onde o usuário está olhando.

5.2.2. Dispositivos de Rastreamento

Os dispositivos de rastreamento ou os *tracking devices* procuram determinar a posição ou a orientação de uma parte do corpo do usuário. Existem seis tipos básicos de rastreadores [ISD 96]:

- a) **mecânicos** - possuem alta velocidade e precisão no rastreamento, em contrapartida, diminuem a mobilidade dos usuários, os quais vestem um capacete que é preso a um braço mecânico articulado;
- b) **ultra-sônicos** - possuem pouca precisão, mas utilizam apenas pequenos emissores de ultra-som presos ao corpo, o que dá bastante mobilidade ao usuário;
- c) **magnéticos** - utilizam conjuntos de bobinas para produzir campos magnéticos e sensores para determinar o tamanho e a direção destes campos;
- d) **extração de imagens** - são acoplados ao corpo do usuário, vários *leds* que devem ser rastreados e filmados por uma câmera, as imagens filmadas são processadas e, em função da posição das luzes, calcula-se a posição do usuário;
- e) **óticos** - podem ser considerados como uma inversão dos rastreadores por extração de imagens. São colocadas quatro câmeras sobre a cabeça do usuário, que filmam o teto que é composto por uma matriz de *leds* estáticos. Através de um padrão que reconhece o caminhar do usuário e registra acendendo ou não os *leds*, pode-se reconhecer o caminhar do usuário e com as outras câmeras sua rotação;
- f) **sem referencial (*sourceless trackers*)** - criados para dar maior mobilidade ao usuário, estes rastreadores possuem três categorias principais: os inclinômetros, as chaves de inclinação e os sensores piezoelétricos de pressão e torção [HOL 95].

5.2.3. Luvas Eletrônicas

As luvas eletrônicas buscam capturar os movimentos das mãos e dos dedos para utilizá-los como forma de interação com o usuário. Os modelos mais conhecidos são: a *Data Glove*, criada pela empresa VPL, que utiliza fibra ótica e mediadores de luminosidade, e a *Power Glove*, criada pela empresa *Mattel*, que usa tinta condutora para aferir o movimento dos dedos.

5.2.4. Dispositivos Geradores de Sensação de Tato e de Força

Estes dispositivos são denominados de *haptic interfaces* e são responsáveis por produzirem a sensação de tato (*touch feedback*) ou de força (*force feedback*). São utilizados em ambientes de RV para enfatizar ainda mais a sensação de imersão.

A sensação de tato provê informações sobre a geometria da superfície, sua textura ou sua temperatura. Por outro lado, a sensação de força fornece informações sobre o peso do objeto e sua consistência.

5.3. Aspectos de um programa de RV

Os componentes básicos de um sistema de RV podem ser divididos em: processador de entrada, processador de simulação, processo de *rendering* e base de dados do mundo [ISD 93]. Todas estas partes devem considerar o tempo requerido para processamento. Cada atraso no tempo de resposta, diminui o sentido de “presença” e realidade da simulação.

5.3.1. O Processo de Entrada

Este processo controla os dispositivos usados para dar as informações de entrada para o computador. Há uma grande variedade de possíveis dispositivos de entrada: *mouse*, teclado, *joystick*, luva, óculos, HMD, etc. Um sistema de RV baseado em rede, deve ainda adicionar as entradas recebidas pela rede.

Na verdade, o processamento de entrada de um sistema RV é bastante simples. O objetivo é dar as coordenadas dos dados ao sistema com o mínimo de tempo gasto.

5.3.2. O Processo de Simulação

Pode-se dizer que o ponto central de um sistema RV é o processo de simulação. Este processo deve conhecer os objetos e as várias entradas. Ele trata as interações, os *scripts* das ações dos objetos, simulações das leis físicas (reais ou imaginárias) e determina o *status* do mundo. Esta simulação é basicamente um processo discreto que é executado uma vez para cada passo ou *frame*. Uma aplicação RV em rede pode ter múltiplas simulações rodando sobre diferentes máquinas, cada uma com a contagem de tempo diferente. A coordenação destas

pode ser uma tarefa uma tanto complexa.

Esta fase é responsável por pegar as entradas dos usuários junto com as tarefas programadas dentro do mundo tais como colisão, detecção, *scripts*, etc., e determinar as ações que irá tomar no mundo virtual.

5.3.3. O Processo de Rendering

Os *rendering processes* criam as sensações levadas ao usuário e são responsáveis pela criação da imagem virtual a partir da descrição do mundo. Uma aplicação RV em rede deveria também ter dados de saída para outros processos da rede. Deve haver processos separados de *rendering* para sistemas de sensoriamento, visão, audição, toque, e outros. Cada processo deveria ter uma descrição do estado do mundo do processo de simulação ou diretamente da base de dados para cada contagem de tempo - *time step*.

5.3.4. A Base de dados do mundo virtual

O armazenamento das informações dos objetos e do mundo é a maior parte do projeto de um sistema de realidade virtual. As primeiras coisas a serem armazenadas na base de dados do mundo ou nos arquivos de descrição do mundo, são os objetos que habitam o mundo, *scripts* que descrevem as ações destes objetos ou os usuários (coisas que podem acontecer ao usuário), iluminação, além dos dispositivos de *hardware*.

É possível armazenar tais informações em um único arquivo, em um conjunto de arquivos ou em uma base de dados. O método de múltiplos arquivos é o mais utilizado, nesse caso a cada objeto associa-se um ou mais arquivos (geometria, *scripts*, etc.) e há ainda alguns arquivos do mundo que acionam outros arquivos para serem carregados. Alguns sistemas também incluem um arquivo de configuração que define as conexões de interfaces de *hardware*.

A base de dados pode ser carregada durante o *startup* do sistema, ou o sistema pode apenas ler os arquivos necessários.

Os dados dos arquivos geralmente são armazenados como arquivos texto (ASCII),

entretanto em muitos sistemas eles são substituídos por arquivos binários. Alguns sistemas possuem toda a informação do mundo virtual compilada dentro da própria aplicação.

5.3.5. O Mundo Virtual

O próprio mundo virtual precisa ser definido como um “espaço do mundo”. O computador deve relacionar um valor numérico sobre as localizações de cada ponto para cada objeto dentro do mundo num sistema de coordenadas, que são geralmente expressas em dimensões cartesianas (X, Y e Z).

A maior limitação sobre o espaço do mundo é o tipo de números usados para as coordenadas, que podem ser coordenadas de ponto flutuante ou coordenadas de ponto fixo. A escolha se baseia na velocidade além do desejo de se ter um campo de coordenada uniforme.

Geralmente o mundo virtual é dividido em múltiplos mundos ou cenários, o que ajuda a tratar com a limitação sobre o espaço de coordenadas do mundo. Neste caso, a aplicação deve prover também um meio de transitar entre os cenários - conhecido como *portals*. Isto ajuda também a minimizar a quantidade de objetos a serem computados tanto para os *scripts* quanto para o *rendering process*.

Em um sistema que suporta múltiplos ambientes a base de dados do mundo necessitaria de múltiplos arquivos de descrição de cena, cada arquivo deveria conter os nomes dos objetos na cena, descrição (por exemplo, tamanho, fundo do cenário, iluminação, etc.).

6. REALIDADE VIRTUAL E O ENSINO

Pode se afirmar que, a realidade virtual tem o potencial para modificar a forma como as pessoas aprendem, pois permite que o aprendiz explore ambientes, processos ou objetos, não através de livros, fotos, filmes, mas através da manipulação e análise virtual do próprio alvo de estudo. Isto leva o aprendiz ao próprio contexto do assunto a ser aprendido, e receba a cada ação que fizer, uma realimentação deste contexto.

A questão que permanece é: como este novo meio pode ser produtivamente incorporado dentro do processo de aprendizagem?

6.1. Como, Quando e Por Que Usar a RV na Educação

Segundo Pantelidis em [PAN 95] existem diversas razões para usar a RV gráfica na educação. São elas:

- a) provém maior motivação nos usuários;
- b) possui um poder de ilustração para alguns processos e objetos muito maior do que outras mídias;
- c) permite tanto uma análise de muito perto quanto de muito longe para os objetos;
- d) permite que pessoas deficientes realizem tarefas que de outra forma não são possíveis;
- e) dá oportunidade para compreensão baseada em novas perspectivas;
- f) permite que o aprendiz desenvolva o trabalho no seu próprio ritmo;
- g) permite ao aprendiz proceder através de um experiência durante um período de tempo não fixado pelo período de aula regular;
- h) requer interação, encorajando a participação ativa do aprendiz.

Ainda a mesma autora, em [PAN 96], apresenta várias sugestões de quando usar e quando não usar a RV na Educação.

Deve-se usar a Realidade Virtual quando:

- a) uma simulação poderia ser usada;
- b) o ensino ou treinamento usando algo real poderia ser inconveniente, perigoso ou impossível;
- c) o erro cometido pelo aprendiz usando algo real poderia ser desmotivador para o aprendiz, prejudicial ao ambiente, ou capaz de causar danos ao equipamento;
- d) um modelo de um ambiente irá ensinar ou treinar tão bem quanto com o equipamento real;
- e) interagir com o modelo é tão ou mais motivador do que interagir com algo real;
- f) experiências compartilhadas de um grupo em um ambiente distribuído são importante;
- g) a experiência de criação em um ambiente ou modelo simulado é importante para o objetivo da aprendizagem;
- h) a visualização da informação é necessária e a sua manipulação e reorganização usando símbolos gráficos pode ser também mais facilmente entendida;
- i) situações de treinamento precisam ser feitas de acordo com a realidade, por exemplo nas experiências práticas sobre condições realísticas;
- j) se necessita tornar o imperceptível, perceptível - por exemplo, utilizar volumes sólidos para ilustrar colisão de idéias em um processo de grupo;
- k) desenvolver ambientes e atividades em grupo, só for possível na forma de mundos gerados por computador;
- l) for necessário ensinar tarefas que envolvem habilidade manual ou movimentos físicos;
- m) o aprendizado se tornar mais interessante e lúdico;
- n) for necessário para dar à pessoas desabilitadas ou incapazes, a oportunidade de fazer experimentos e atividades que elas não poderiam fazer.

Não se deve usar a Realidade Virtual quando:

- a) não existe um substituto para o ensino ou treinamento com algo real;
- b) a interação presencial, com professores ou alunos, se fizer necessária;

- c) usar um ambiente virtual puder ser fisicamente ou emocionalmente prejudicial;
- d) usar um ambiente virtual puder resultar numa “literalização” - uma simulação tão convincente que alguns usuários poderiam confundir o modelo com a realidade;
- e) o custo estimado para sua implantação for tão caro, que não justifique o seu uso, considerando o aprendizado esperado.

6.2. Experiências do Uso de Realidade Virtual Aplicada à Educação

O potencial das aplicações educacionais de RV foi categorizado por Rory Stuart e John Thomas da NYNEX no artigo “*The Implications of Education in Cyberspace*”, escrito para *Multimedia Review* no início dos anos noventa. De acordo com Stuart e Thomas, as aplicações RV podem capacitar os aprendizes para:

- a) explorar lugares e coisas existentes nos quais os estudantes não poderiam ter acesso de outro modo;
- b) explorar coisas reais que, sem alterações da escala no tamanho e tempo, não poderiam ser efetivamente examinadas;
- c) criar lugares e coisas com qualidade natural ou alterada;
- d) interagir com pessoas que estão em locais remotos;
- e) interagir com pessoas de modo não-realístico;
- f) interagir com seres virtuais, tais como representações de figuras históricas;
- g) criar e manipular representações conceituais abstratas, como estrutura de dados e funções matemáticas.

Em [HEL 96], Helsel apresenta vários projetos que estão sendo desenvolvidos dentro destas categorias classificadas por Rory Stuart e John Thomas. Um projeto que apresenta vários aspectos semelhantes à dissertação em questão é o chamado *ExploreNet*, apresentado a seguir.

Outras aplicações também são citadas apenas com o intuito de formar um sucinto levantamento acerca dos projetos desenvolvidos ou em desenvolvimento que aplicam RV à Educação. Seria um tanto tedioso e quase impossível querer englobar todas as aplicações existentes.

6.2.1. ExploreNet

Este projeto vem sendo desenvolvido pela *University of Central Florida*, pelo *Computer Science Department* e pelo *Institute for Simulation and Training*, desde 1991, e hoje já conta com três protótipos do sistema em uso e um quarto em desenvolvimento.

ExploreNet é um ambiente gráfico 2D, distribuído, orientado a objetos, com o propósito de produzir ambientes RPGs para o aprendizado cooperativo.

Foi projetado como uma aplicação para *Windows*, e segue a maioria das convenções deste tipo de aplicação, é cliente/servidor, utiliza de arquivos *bitmap* (arquivos com extensão .bmp) para gravar as imagens das cenas e dos personagens do mundo virtual. Como técnicas de comunicação, o *ExploreNet* utiliza *WinSock*, um recurso de comunicação de domínio público, para acessar comunicações Internet. Sua versão 3 foi escrita em *Smalltalk/Vwin32*.

O *ExploreNet* já foi utilizado em projetos com escolas onde os próprios alunos criaram seus *adventures games* interativos e multiparticipantes [MOS 95]. Vários outros projetos já foram e estão sendo desenvolvidos com o *ExploreNet*, para tanto, consulte o endereço eletrônico <http://longwood.cs.ucf.edu/ExploreNet/>.

6.2.2. VETL - Virtual Environment Technology Laboratory

Este laboratório é uma criação em conjunto entre as universidades de *George Mason* e *Houston-Downtown* com a NASA, através dos pesquisadores Chris Dede e R. Bowen Loftin e John Space. Está sendo desenvolvido desde 1994, o projeto *ScienceSpace* [DED 96], que hoje conta com três mundos virtuais que permitem investigar o efeito de ambientes imersivos e multisensoriais no aprendizado da física

O primeiro deles - *NewtonWorld* [DED 94], trabalha com conceitos de mecânica newtoniana como inércia, energia cinética e leis de movimento, neste o aluno é colocado em um corredor, sobre um chão quadriculado, cercado por colunas dos lados e por paredes nas extremidades. Ali é possível pegar e lançar bolas virtuais para qualquer direção. As bolas, enquanto viajam pelo espaço, podem colidir umas com as outras e mudar sua direção.

A idéia do chão quadriculado e das colunas (igualmente espaçadas) é permitir, mesmo que de forma aproximada, que o aluno realize medições no ambiente. Para dar mais realismo às cenas, são usados fones de ouvido nos quais são reproduzidos os sons das colisões e do movimento dos objetos. Através de uma espécie de controle-remoto o aluno pode posicionar-se em qualquer parte do ambiente para melhor observar uma experiência.

O segundo, o *MaxellWorld*, trabalha com a exploração da eletrostática, através das Leis de Gauss.

Partindo de resultados colhidos com o primeiro projeto, o *MaxwellWorld* teve algumas características da interface modificadas. A principal mudança ocorreu com o controle-remoto que passou a ficar preso ao pulso do usuário como se fosse um relógio. Isto permitiu que as mãos do estudante ficassem livres para outras interações. Para navegar, por exemplo, o aluno seleciona a opção no relógio e a seguir aponta com o indicador a direção para onde quer ir.

Usando também a mão o usuário pode, “tocando os objetos”, carregá-los com cargas elétricas e observar seu comportamento. Para melhor compreender os fenômenos é possível medir cargas e exibir superfícies de equipotencial e linhas de fluxo elétrico.

Por último há o ambiente *PaulingWorld*, que permite o estudo de estrutura moleculares através de uma variedade de representações. Este projeto tem sido utilizado tanto como ferramenta de ensino quanto de pesquisa.

Estes mundos virtuais foram construídos através de dois softwares da NASA, que fazem uso de estações gráficas de alta resolução de som e imagem, além de contar com equipamentos *HMD*, *mouse 3D*, entre outros.

6.2.3. Educação Especial

Na *University of Nottingham* - Inglaterra, psicólogos do *Department of Learning Disabilities*, juntamente com um grupo de desenvolvedores de aplicações de RV - *VIRART*, estão trabalhando no desenvolvimento de ambientes educacionais virtuais (VE - virtual environments) para estudantes com problemas de aprendizado, além de realizarem uma série

de experimentos para testar sua validade [CRO 95].

O grupo de estudantes com o qual está sendo desenvolvido tal pesquisa é muito heterogêneo, sendo composto por alunos com retardo mental, síndrome de Down, ou com problemas de mobilidade e cardíacos. Alguns não falam e precisam de sinais para se comunicar. Mas a maioria não tem um diagnóstico médico mais preciso do que apenas “deficiência na aprendizagem”.

O ambiente educacional utilizado, foi desenvolvido com o pacote *Superscape* da *Dimension International*. Eles classificam-se em três grupos:

- a) uma série de *VE* para auxiliar no ensino de *Makaton*, uma linguagem de sinais usada na *United Kingdom*, por pessoas com problemas de aprendizado;
- b) uma série de *VE* experimental (por exemplo, uma casa, uma cidade) destinados a tornar disponível aos estudantes experiências nas quais eles nunca poderiam ter na vida real;
- c) *VE* para ensinar habilidades específicas do dia-a-dia e facilitar pesquisa empírica: um supermercado virtual, um modelo virtual de escola, etc.

Tais ambientes são construídos em trabalho conjunto com psicólogos e programadores em todas as etapas do processo de construção.

Várias análises já foram desenvolvidas sobre este projeto, e através destas, os pesquisadores já conseguiram concluir a importância dos *VE* para ensinar tarefas do cotidiano aos alunos portadores de tais deficiências, além de observarem a importância da *RV* na promoção de atividades auto-dirigidas pelos estudantes.

Os pesquisadores têm então desenvolvido estudos para definir a base teórica deste trabalho, que integre a metodologia pesquisada e o projeto dos ambientes virtuais com conceitos e técnicas comuns à educação especial.

6.2.4. VESAMOTEX - Virtual Education Science and Math of Texas

O projeto VESAMOTEX está sendo desenvolvido em *Slaton High School* - em Slaton

no Texas [TAL 96], e tem quatro fases distintas:

- a) **primeira fase** - inclui a investigação e compra de equipamento básico;
- b) **segunda fase** - envolve os estudantes e professores na produção de várias aplicações matemáticas e científicas através da produção de vídeo e de outras mídias;
- c) **terceira fase** - envolve todos os estudantes na utilização de RV nas aulas de ciências, matemáticas e computação;
- d) **quarta fase** - envolve a demonstração de RV para escolas dentro do mesmo distrito e áreas vizinhas.

Os softwares utilizados neste projeto incluem *3D Studio*, *AutoCad*, *Simply 3D*, vários programas shareware, além de programas específicos de RV como *Virtus WalkThrough Pro*, *VREAM (Virtual Reality Development System)* e *VR Creator*, além do *RenderWare* e *Superscape*.

O projeto iniciou em meados de janeiro de 1996, e os professores e estudantes estão aprendendo como usar o *software* juntos. Atualmente os estudantes estão envolvidos com a RV através da Internet e além do desenvolvimento de projetos relacionados aos seus estudos em eletromagnetismo através de um modelo atômico.

Recentemente foi elaborado um estudo entre os estudantes, para obter informações sobre o impacto do programa. Depois de três semanas com trabalhos introdutórios sobre RV, os alunos foram questionados. Os resultados já conseguiram mostrar o interesse dos alunos em trabalhar com realidade virtual, outras conclusões e novas observações e testes estão sendo elaborados.

6.2.5. O Ensino de Literatura

Juanita K. Stock em [STO 96], discute a utilização de RV no ensino de literatura, e apresenta sua experiência com uma turma de alunos em *Haywood Community College*.

Com a ajuda de professores de informática da escola, foi criado um mundo virtual imersivo, com a cena do livro "*A good man is hard to find*" de Flennerly O'Conner. Com a

utilização de óculos de RV, os alunos eram imersos na cena e podiam caminhar livremente pela estória.

Através desta experiência, os alunos que já haviam lido o livro, ao reconhecerem os personagens no mundo virtual passaram a discutir o texto e analisar mais detalhes da estória, além de comparar o mundo virtual construído com a sua própria interpretação do livro. Alguns alunos motivaram-se inclusive a reler o texto a fim de identificar melhor os personagens. Outros, porém, que ainda não haviam lido a obra, se sentiram motivados a fazê-lo para adquirir uma maior compreensão sobre o “livro virtual”.

Os pesquisadores envolvidos no projeto ficaram muito entusiasmados com os resultados e iniciaram uma nova fase do projeto, eles estão capacitando os alunos para que criem seus próprios “livros virtuais” a partir de sua própria visão da obra literária. A idéia básica deste projeto, é fazer com que os alunos ao lerem um texto sintam-se motivados a prestar atenção nos detalhes e procurem criar mentalmente as cenas descritas nos livros.

6.3. Fontes de Pesquisa

Pode-se citar como grandes fontes de pesquisa sobre a utilização de RV aplicada à educação, o “*Virtual Reality and Education Laboratory - VREL*” da *East Carolina University*, através do endereço eletrônico <http://eastnet.educ.ecu.edu/vr/>; o “*Human Interface Technology Laboratory*” da *University of Washington*, via endereço eletrônico <http://www.hitl.washington.edu/>; além de revistas como a *VR in the Schools* (publicada pelo VREL), a *RV News* (email: 100024.1425@compuserve.com), *CyberEdge Journal* (email: bdel@well.sf.ca.us), listas de discussão, entre outras. Em [ISD 93] pode-se encontrar várias referências para pesquisa em RV.

7. SISTEMA DE AUTORIA

"Creating constructivist, collaborative educational experiences that encourage the former outcome is an important means for ensuring full public access to and control of this emerging meta-medium." [DED 95]

Este capítulo apresenta a especificação de um sistema de autoria para a criação de *adventures educacionais*. Tal sistema pretende ser um facilitador na criação de *adventures educacionais*, a ponto de que os próprios professores e alunos se tornem autores de jogos educativos.

A ferramenta em si permite ao autor a criação de objetos virtuais e também a disposição destes objetos, constituindo assim os cenários do ambiente. Os objetos podem ser de diversas formas, desde cadeiras, lixeiras, computadores, e, também, personagens que irão compor o micro-mundo. Nesta perspectiva, o próprio aprendiz inserido no ambiente do *adventure educacional* será modelado como um de seus personagens.

A ferramenta também permite a criação de duas formas de interação dentro do ambiente. Através de verbos que irão ativar os objetos, podendo alterar ou não seu comportamento, e através da comunicação entre personagens, sejam eles "reais" ou não - os quais seriam representados pelos aprendizes e pelos "bots", respectivamente. A conversação entre personagens é feita através de linguagem natural, podendo induzir certas respostas e/ou mudança de comportamento por parte do outro personagem.

O sistema possui alguns componentes básicos:

- a) ambiente físico virtual, composto pelos personagens (representação dos usuários) e pelos objetos, os quais são orientados a eventos;
- b) conjunto de agentes autônomos ("bots"), que ajudam a povoar o mundo;
- c) interface de usuário, que permite um ou mais usuários participarem e interagirem no mundo;
- d) módulo de processamento de linguagem natural, que permite a interação entre usuários e *bots*.

A prototipação do sistema foi desenvolvida com a linguagem Delphi 2.0. Este protótipo serve apenas como modelo para a implementação final do sistema de autoria aqui proposto.

Os objetos tridimensionais são modelados através da linguagem VRML 2.0 e seus *scripts* em Java.

7.1. ESPECIFICAÇÃO DO SISTEMA

O sistema de autoria proposto engloba vários componentes, como foi descrito na seção anterior. Na dissertação em questão não se pretendia ter um sistema de autoria com este porte inacabado, fato que seria um tanto quanto doloso, devido é lógico, ao tempo que se dispunha e à complexidade do projeto.

A ênfase do esforço neste projeto foi então buscar especificar um sistema que elucidasse o melhor de um ambiente de aprendizado e, a partir daí, foi desenvolvido um componente do sistema, que trata principalmente da interação com os objetos e da sua criação, assim como, dos cenários e, por conseguinte, da elaboração do micro-mundo.

O sistema desenvolvido possui três módulos principais, que estão intimamente ligados, e três bibliotecas: de cenários, de objetos pré-definidos e de personagens. Pode-se dizer, pelo modo que o mesmo foi implementado, que o módulo de composição engloba os sub-módulos edição e implementação. A relação entre eles é apresentada na Figura 7.1 e foi criada como se fossem módulos separados para melhor visualizar a integração dos módulos com as bibliotecas.

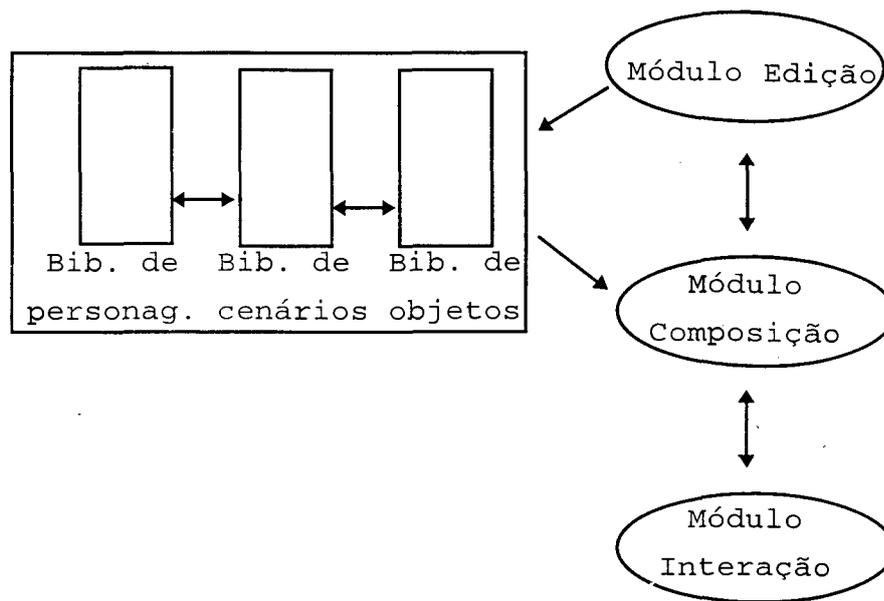


Figura 7.1: Inter-relacionamento entre os módulos e bibliotecas do sistema

7.1.1. Módulo Edição

O módulo de edição é um módulo composto por ferramentas de edição, onde o autor pode criar imagens tridimensionais para utilizar em seu micro-mundo. É o módulo de criação dos cenários, objetos e personagens. Nele o autor pode também importar imagens feitas em outros editores de imagens 3D (em VRML), ou contar com alguns filtros, também chamados de conversores, que transferem outros tipos de arquivos para arquivos VRML 2.0, à exemplo do conversor Community Place criado pela Sony.

Tanto as imagens criadas neste módulo, como também as importadas, podem ser armazenadas nas bibliotecas do sistema, criando assim bibliotecas cada vez mais personalizadas além de facilitar a reutilização das imagens.

7.1.2. Módulo Composição

O módulo de composição, como o próprio nome diz, permite a composição do micro-mundo através da elaboração dos cenários, definindo a disposição dos objetos em cada ambiente que compõem o mundo virtual.

Pode-se dizer que este é o módulo principal, pois neste módulo o autor seleciona as imagens dos cenários, dos objetos e dos personagens (através das respectivas bibliotecas) e também

define as variáveis de cada objeto. Por exemplo o objeto torno mecânico teria o seguinte conjunto de variáveis: ligado (indica se o aparelho está ligado ou não), velocidade de rotação (indica a velocidade de rotação do aparelho), profundidade (indica o quanto a ferramenta retira de material da peça), etc. Este conjunto de variáveis representaria o estado do objeto.

Outra característica importante dos objetos do micro-mundo é que cada um deles pode ter diversas imagens associadas. Utilizando o exemplo do torno mecânico, poderia haver uma imagem associada ao seu estado desligado e outra ao estado ligado. Nesse caso o botão *on* estaria aceso, e haveria um vínculo a um som que representaria o funcionamento do aparelho, etc.

O autor quando utiliza da biblioteca de objetos pré-definidos, na verdade, está selecionando imagens que estarão associadas a um nome (nome do objeto). A partir daí, ele define as variáveis do objeto e ativa o módulo de interação.

7.1.3. Módulo de Interação

Neste módulo o autor define as regras de interação com os objetos e quais alterações estas regras causarão em cada objeto, as formas de diálogo entre os personagens, além de definir como os cenários estão ligados uns aos outros.

A interação com os objetos é criada a partir de vários comandos verbalizados, estes comandos definem juntamente com o conjunto de variáveis do objeto, o comportamento que este terá ao ser acionado. O sistema de autoria possui um conjunto de comandos e o autor define a partir destes, qual comportamento terá cada objeto em relação a cada comando. Estes comandos ficam disponíveis no micro-mundo em forma de botões que permitem ao aprendiz manipular os objetos. São eles:

- a) abrir - permite abrir um objeto para examinar seu interior, por exemplo, abrir uma caixa de correspondências;
- b) fechar - esta opção serve para fechar um objeto aberto, por exemplo, fechar uma caixa ou mesmo fechar uma porta;
- c) pegar - permite guardar o objeto selecionado com esta opção no *container* do

usuário, por exemplo, armazenar uma chave que depois pode ser usada para abrir uma porta;

- d) entregar - esta opção é utilizada quando se deseja entregar algum objeto que está em posse do usuário para outro personagem, ou mesmo largá-lo no ambiente;
- e) usar - com esta opção pode-se utilizar um objeto que está de posse do usuário ou não, ou seja, o usuário pode utilizar um objeto que está no *container* ou que está diretamente do ambiente;
- f) examinar - permite examinar com mais detalhes o objeto selecionado, podendo, por exemplo, analisar o estado interno do objeto;
- g) mover - este comando é utilizado para mover um objeto no ambiente.

Estes comandos são representados pelo sistema em forma de botões que compõem um menu fixo na parte inferior da tela, portanto o aprendiz pode caminhar e explorar livremente o ambiente, inspecionando os objetos de ambos os lados. A interação geralmente ocorre através de um clique no botão seguido de um clique no objeto que se deseja manipular.

O menu também possui um botão representando o *container* do usuário, que ao ser acionado mostra os objetos em seu poder. Mesmo dentro do *container*, os objetos podem ser manipulados através dos comandos.

As operações de inspeção e também de navegação pelo ambiente são realizadas através do *browser* 3D, que já implementa as possibilidades de caminhar, flutuar dentro do ambiente, além de rotacionar um objeto, aproximar-se ou distanciar-se dele.

Os tipos de comportamentos que o objeto pode ter em relação a cada comando acionado, são listados abaixo:

- a) responder com uma frase;
- b) ser retirado do cenário, neste caso tem-se três hipóteses:
 - i. ser retirado do objeto do ambiente e guardá-lo com o usuário, (no caso, o sistema implementa um *container* (repositório) para cada usuário, onde este

- pode armazenar objetos e utilizá-los quando desejar);
- ii. ser pego por um usuário e entregue a outro;
 - iii. ser pego por um usuário para ser utilizado (por exemplo, uma chave mecânica é retirada de cima do balcão e colocada no torno mecânico);
- d) acionar um outro cenário (por exemplo, uma porta fechada, ao ser acionado o comando abrir, seria carregado um outro cenário), neste caso, a relação objeto-comando seria um *link* para um outro cenário;
- e) ter o seu estado testado ou testar de algum outro, ou seja, testar o valor de alguma variável, como por exemplo, verificar qual imagem de um determinado objeto está sendo visualizada no momento;
- f) mudar seu estado, ou o estado de outro objeto, ou seja, alterar alguma de suas variáveis;
- g) não responder , ou seja o comando nada interfere no comportamento do objeto.

O autor do micro-mundo pode definir os comportamentos dos objetos, de dois modos, através do Modo Visual e Modo *Script*. Estes objetos são denominados de orientados à eventos, pois a cada evento, o objeto pode responder de uma maneira diferente.

No Modo Visual o autor conta com as ações definidas e trabalha a partir da seleção das ações, através de botões. Ele pode também compor várias ações que o objeto executará em função da ocorrência de um evento apenas. Por exemplo, o objeto pode ao mesmo tempo responder com uma frase e alterar alguma de suas variáveis. No sistema isto é chamado de múltiplos comportamentos.

O Modo Visual é propício para pessoas que não possuem conhecimento sobre programação, facilitando a definição dos comportamentos para os objetos sem precisar aprender uma linguagem de programação.

Para tanto, o Modo *Script* é propício a programadores, pois neste modo o autor define os comportamentos através da linguagem Java.

O autor pode também desenvolver no Modo Visual e conferir sua programação no Modo *Script*.

Após executar o módulo de interação, o objeto criado é exibido nas coordenadas $X=1$ e $Y=1$ do cenário, podendo então ser arrastado para a posição desejada.

O módulo de interação para os personagens não foi implementado, pois engloba a teoria de agentes além de um módulo de tratamento de linguagem natural, que não são tratados neste trabalho.

7.1.4. Bibliotecas

O sistema conta com três bibliotecas que possuem imagens de cenários, de objetos pré-definidos e de personagens. Estas imagens podem ser utilizadas, reutilizadas e também modificadas pelo autor, criando assim outras imagens a serem adicionadas às bibliotecas, tornando-as cada vez mais personalizadas.

- a) biblioteca de cenários - disponibiliza alguns modelos de fundo de cenário, tais como um deserto, um dia ensolarado, um sala (vazia), entre outros;
- b) biblioteca de objetos pré-definidos - possui diversos tipos de imagens de objetos, por exemplo, caixa de correio, cadeira, aparelhos, livros, etc. Está dividida em repositórios, cada um possui objetos afins, por exemplo, repositório de móveis, lá se encontram cadeiras, mesas, estantes, entre outros;
- c) biblioteca de personagens - contém diversos tipos de personagens, que representam os “seres humanos”, como também os “bots”. Está também dividida em repositórios.

7.1.5. Sistema de Arquivos

A ferramenta de autoria ao ser instalada é armazenada obedecendo uma árvore de diretórios. O diretório principal possui a aplicação propriamente dita (arquivo com extensão .exe), e três sub-diretórios (arquivos com extensão .bmp e .wrl). O primeiro diretório contém os arquivos relacionados às imagens dos cenários; o segundo possui os arquivos

correspondentes às imagens dos objetos; e por fim, o diretório que armazena as imagens dos personagens. Veja Figura 7.2.

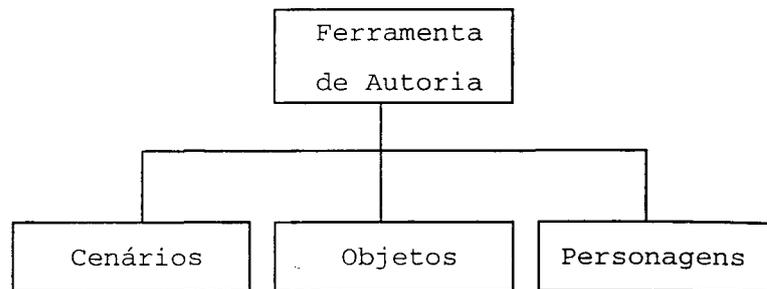


Figura 7.2: Árvore de diretórios da ferramenta de autoria

7.2. CONSTITUIÇÃO DOS “ADVENTURES EDUCACIONAIS”

Os *Adventures Educacionais* criados pelo sistema de autoria proposto são denominados de micro-mundos. Os micro-mundos são implementados em um ambiente de realidade virtual distribuído, tridimensional, composto por vários cenários, cada um dos quais contendo *objetos* com propriedades próprias e *personagens*. Os cenários podem estar ligados uns aos outros de diversas maneiras, o que é definido pelo autor do micro-mundo. Em tal ambiente, o caminho que o usuário percorre através de suas próprias ações não ocorre de forma linear, ou seja, não há de forma explícita uma ordem seqüencial dos cenários. A Figura 7.3 busca ilustrar tal conceito.

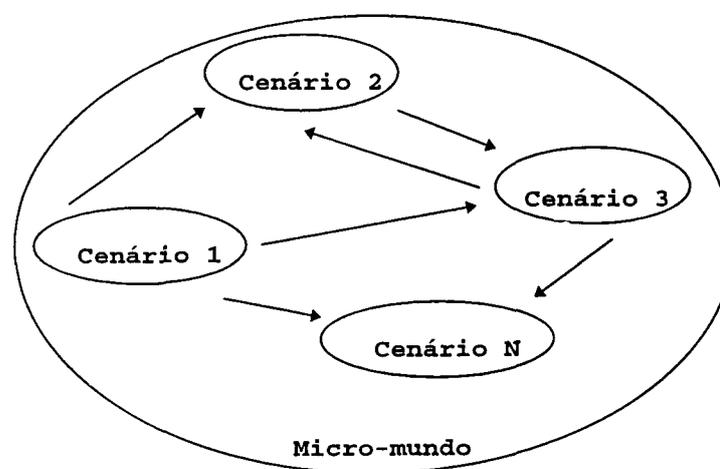


Figura 7.3: Figura ilustrativa sobre a constituição dos micro-mundos

Os objetos são tridimensionais e possuem propriedades tais como forma, tamanho, cor, textura, etc., possuem também um conjunto de variáveis próprias que determinam o seu estado, além de um comportamento orientado a eventos. Este comportamento é modificado em função do evento gerado (ou seja, através da ação executada sobre ele), e do seu estado.

Os personagens são capazes de manipular os objetos do micro-mundo, podendo eventualmente carregá-los e mudar seus estados, além de interagir uns com os outros através de diálogos em linguagem natural. Os personagens podem ser *usuários (aprendizes)* ou *programas ("bots")*, ambos tendo representações físicas semelhantes dentro do micro-mundo.

Cada "bot" representa um papel no micro-mundo. Em contato com os aprendizes, o "bot" pode atuar como uma espécie de tutor, corrigindo rumos, dando sugestões, forçando o aprendiz a modificar sua ação, etc., dependendo da sua personalidade. Para implementar tal conceito se torna propício utilizar da teoria de agentes, [RUS 95, WOO 95].

7.2.1. Características dos Micro-Mundos

As principais características desejadas nos micro-mundos criados a partir da ferramenta especificada são descritas a seguir:

- a) multi-usuário - permite mais de um usuário estar participando do ambiente ao mesmo tempo;
- b) distribuído - permite usuários em locais geograficamente distantes estarem conectados ao ambiente;
- c) modificabilidade e interação em tempo-real - os usuários podem fazer modificações no ambiente, e estas modificações devem ser vistas por todos os usuários, com uma velocidade tal que aproxime-se da realidade, assim também ocorre com a interação;
- d) ambiente de realidade virtual não imersiva - dispensa o uso de dispositivos de realidade virtual, possui uma representação física para cada usuário dentro do ambiente - denominada de *avatar*;
- e) representação gráfica e som tridimensional - os objetos e personagens do ambiente estão representados em 3D, o som tridimensional cria uma sensação mais realística

de som, aumentado e diminuindo o volume ao se aproximar e afastar do emissor de som;

- f) interação através de linguagem natural - os personagens do ambiente se comunicam através da digitação de frases em linguagem natural;
- g) interação através de comandos - os personagens interagem com os objetos através da ativação de comandos.

7.2.2. Sistema de Arquivos dos Micro-Mundos

Os micro-mundos construídos através da ferramenta de autoria, são armazenados através do método de múltiplos arquivos, ou seja, a aplicação toda não constitui apenas de um arquivo.

A aplicação possui arquivos para a descrição dos cenários, os quais devem conter os nomes dos objetos e dos personagens na cena além de alguns dados sobre o cenário, por exemplo, tamanho, fundo do cenário, iluminação, etc.

Possui também arquivos para cada objeto - arquivo de propriedades, arquivo de scripts, etc. - além do arquivo principal, o qual possui a estrutura do mundo e é o responsável por acionar os outros arquivos.

8. APRESENTAÇÃO DA FERRAMENTA

Ao entrar no ambiente o autor inicia nomeando o micro-mundo que deseja criar. Veja Figura 8.1. O botão “ler de arquivo” é utilizado quando se deseja abrir um micro-mundo já existente.

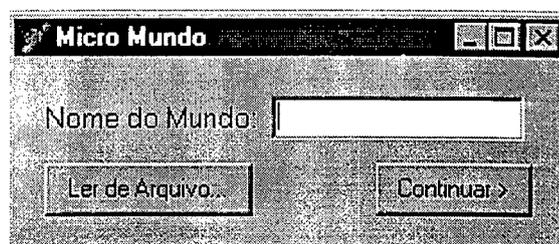


Figura 8.1: tela inicial da ferramenta

Após nomear o mundo o autor deve escolher o tipo de cenário que ele deseja construir. Os estilos de cenários estão disponíveis na biblioteca de cenários (armazenados em forma de árvore de diretório). Ao clicar em um determinado cenário, sua imagem é exibida em uma caixa de diálogo afim de auxiliar na escolha do cenário, conforme mostra a Figura 8.2.

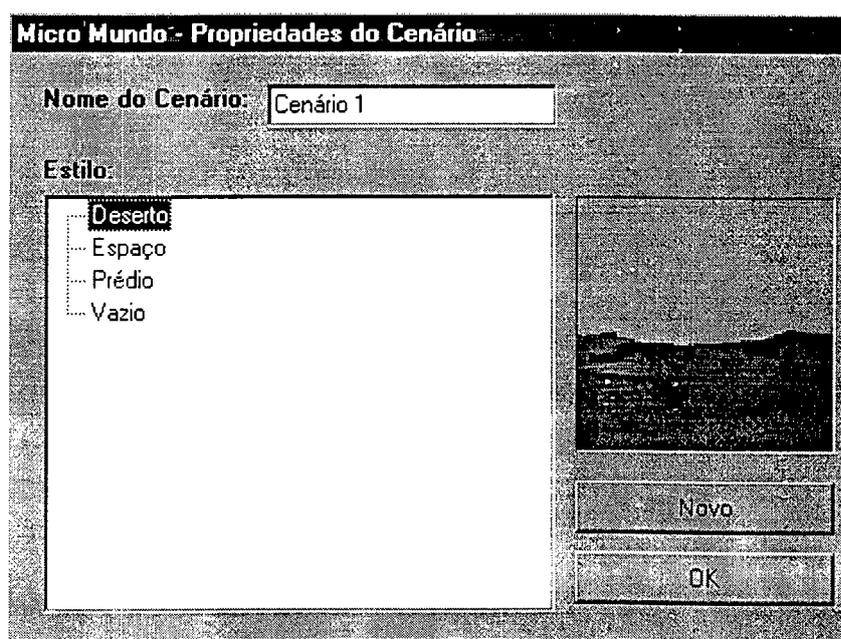


Figura 8.2: Tela Propriedades do Cenário

Através do botão “novo” ativa-se o **Módulo de Edição**. Este módulo possibilita ao

autor criar um novo cenário ou modificar um já existente na biblioteca, entretanto, este módulo não está disponível para esta versão do protótipo.

Após definir o cenário, é carregada a janela principal da ferramenta a qual corresponde ao **Módulo de Composição**. Esta janela está formada como segue: Cenários, Objetos e Personagens, Campo de Visualização, Barra de Ferramentas Personalizada, e Coordenadas (X,Y) da posição do mouse para o campo de visualização. Veja a Figura 8.3.

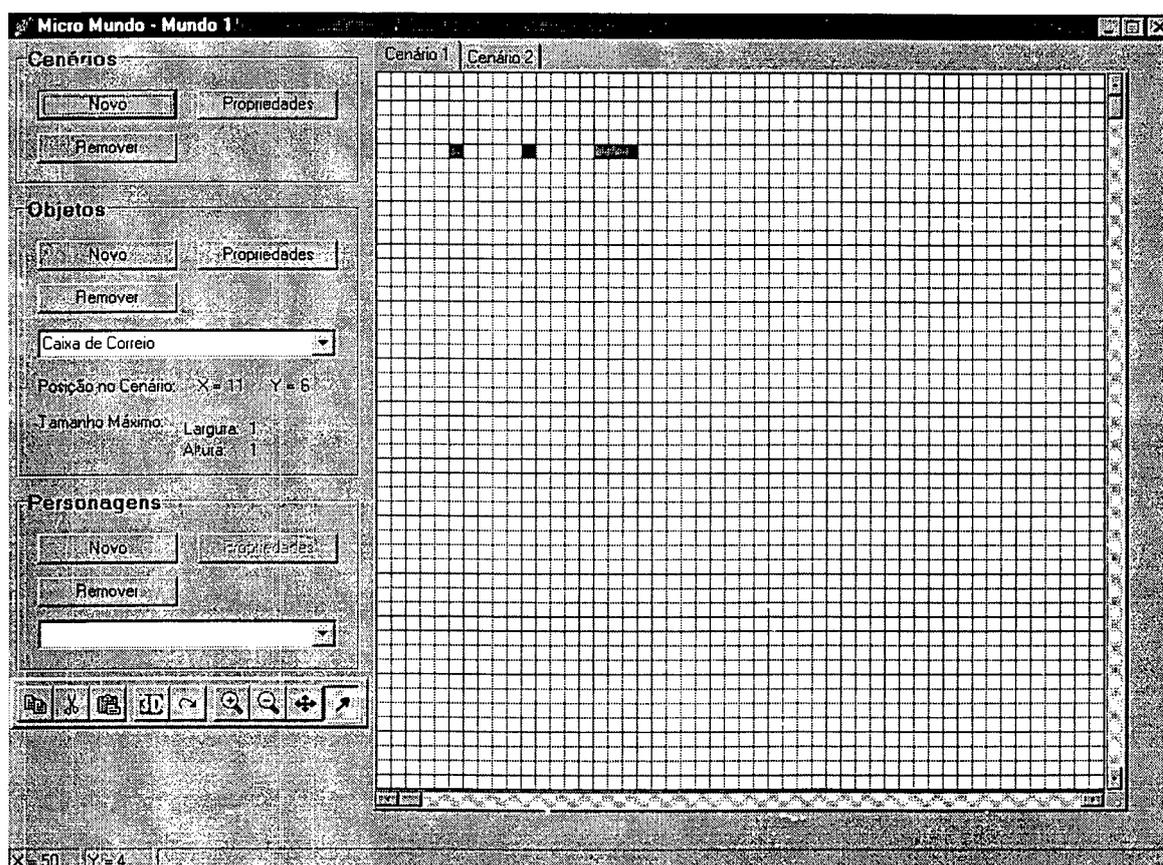


Figura 8.3: Tela Principal da Ferramenta de Autoria

Na partição Cenários o autor pode através do botão “novo” criar novos cenários, sendo que estes são automaticamente indexados um após o outro, como uma seqüência de fichas. O cenário ativo é sempre a “ficha” superior - na Figura 8.3 é o Cenário 1. O botão “propriedades” permite mudar a propriedade do cenário ativo, enquanto o botão “remover” automaticamente exclui o cenário ativo.

A janela correspondente aos botões “novo” e “propriedades” é aquela apresentada na

Figura 8.2.

Na partição Objetos o autor pode criar os objetos a partir do botão “novo” que ativa a janela Propriedades do Objeto, a qual está dividida em três guias principais: Dados Gerais, Variáveis do Objeto e Módulo de Interação (Figura 8.4).

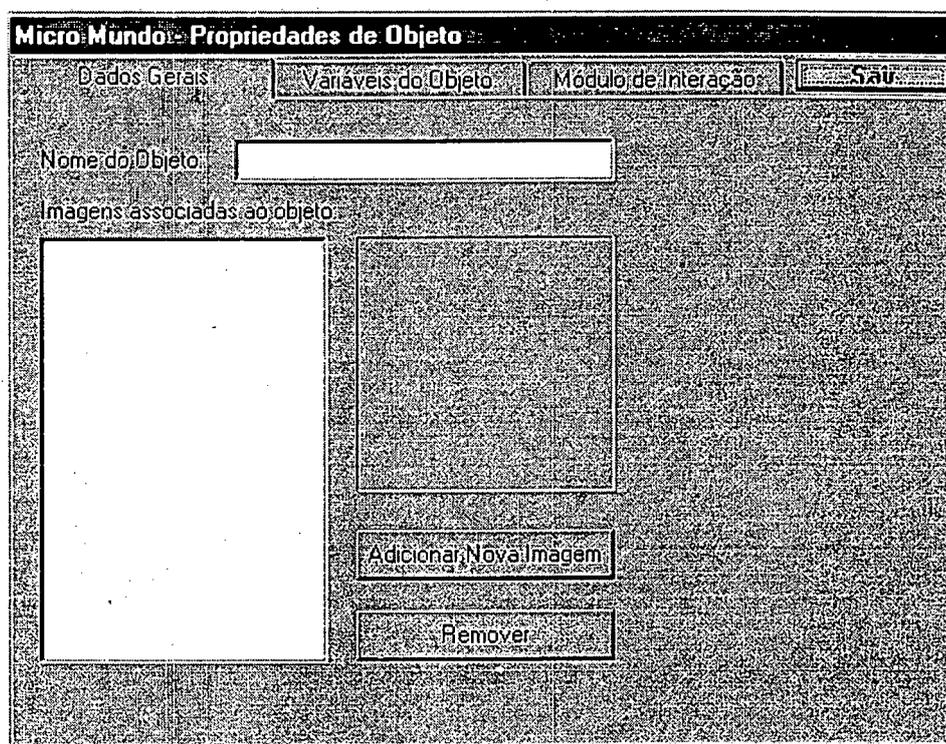


Figura 8.4: Tela Propriedades do Objeto

Na guia Dados Gerais, através do botão “adicionar nova imagem”, o autor pode selecionar as imagens para criar o objeto, conforme Figura 8.5.

O autor pode relacionar quantas imagens quiser para um dado objeto. As imagens selecionadas são então mostradas na guia de Dados Gerais. A imagem que ficar selecionada será a primeira a ser exibida. No exemplo da Figura 8.6, o objeto criado chama-se porta e possui duas imagens relacionadas - porta aberta e porta fechada.

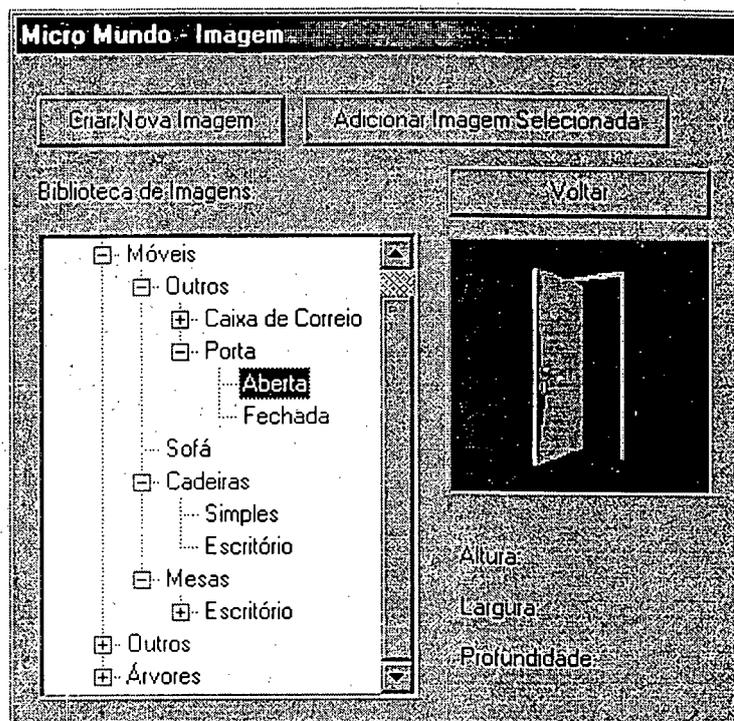


Figura 8.5 : Biblioteca de Imagens dos Objetos

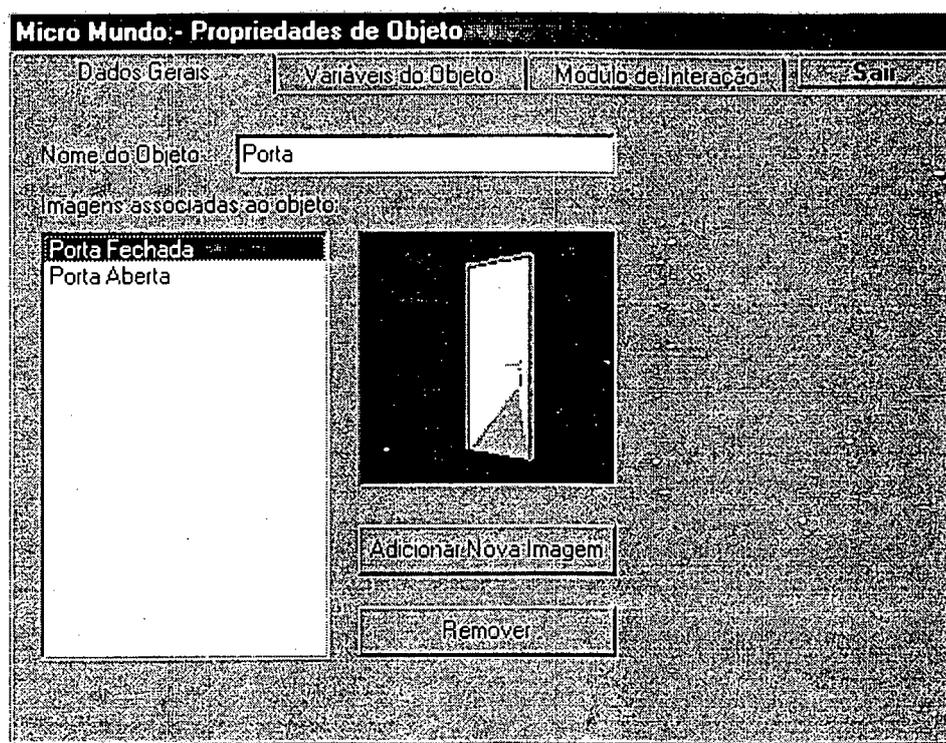


Figura 8.6 : Tela Dados Gerais do Objeto Porta

Na guia Variáveis do Objeto, o autor define as variáveis relacionadas ao objeto que está sendo criado, e também atribui um valor inicial à estas. A variável “imagem” é criada automaticamente pelo sistema e contém a identificação da imagem a ser exibida (ordem seqüencial das imagens selecionadas a partir do zero). As outras variáveis (trancada e altura) foram criadas para enriquecer o exemplo. As variáveis criadas pelo sistema podem ser do tipo: inteira, real, texto e booleana. Veja Figura 8.7.

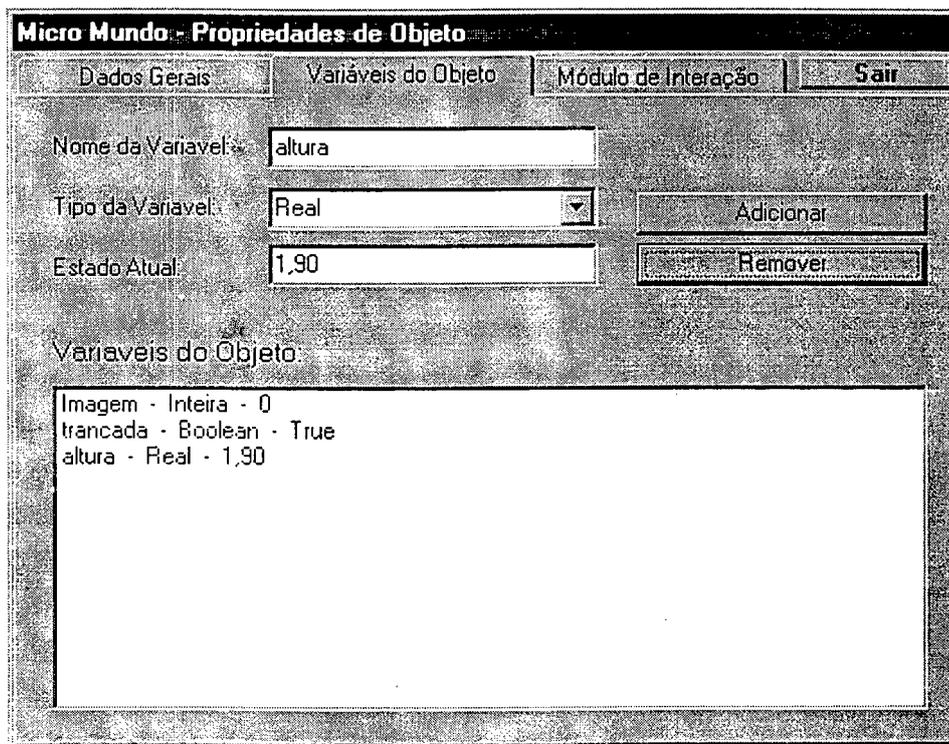


Figura 8.7: Tela Variáveis do Objeto Porta

Na guia **Módulo de Interação**, o autor define através do Modo Visual ou do Modo *Script*, os diversos comportamentos que o objeto terá em função da lista de ações disponíveis.

As figuras 8.8 à 8.11 mostram um exemplo onde se está definindo o comportamento para a ação “abrir” do objeto porta. Na Figura 8.8, o autor escolheu o comportamento “fazer um teste” com o próprio objeto. Através do botão “condição” é ativada a janela da Figura 8.9, onde o autor pode criar a condição a ser testada. No exemplo a condição é “se a variável trancada do objeto porta está setada para true”.

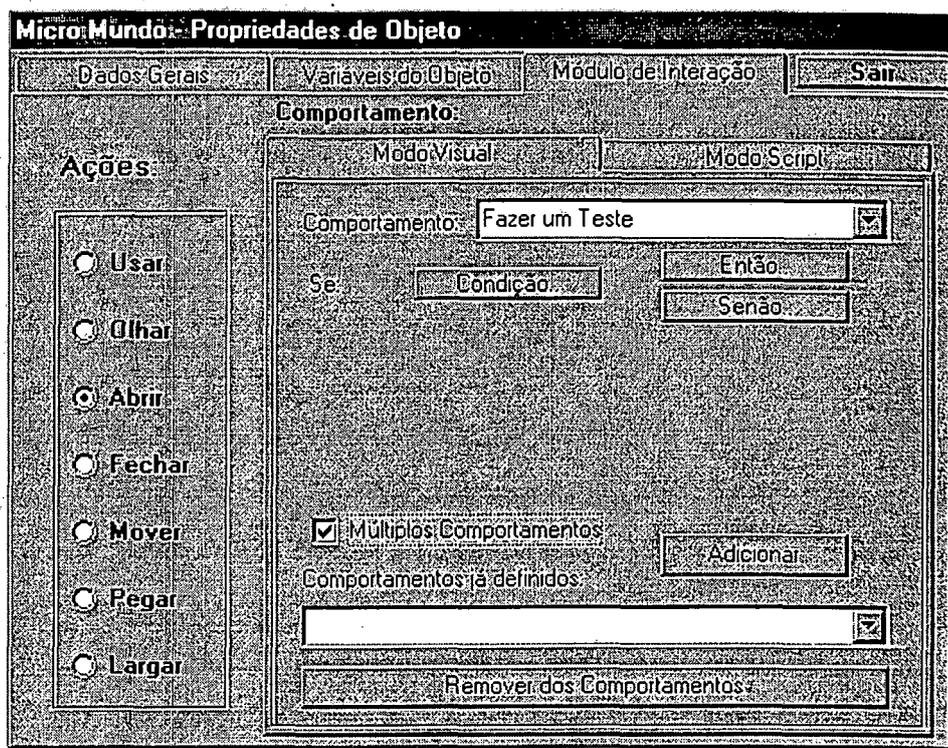


Figura 8.8: Tela do Módulo de Interação - Modo Visual

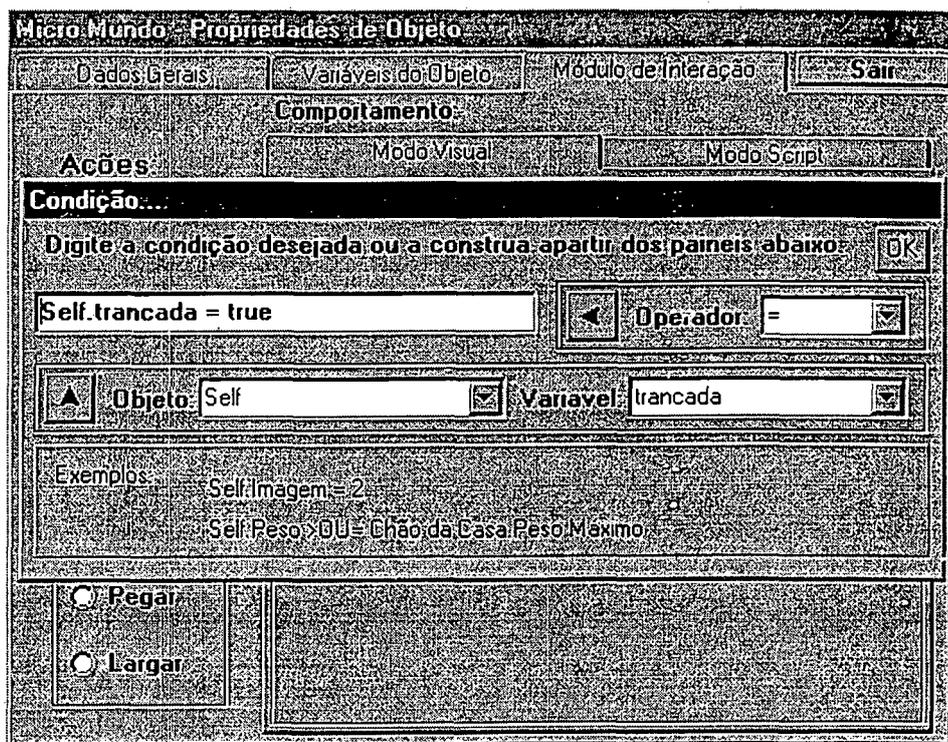


Figura 8.9: Tela de elaboração do teste condicional “se-então”

A próxima etapa consiste na definição dos comportamentos caso a condição seja verdadeira, através do botão “então”, e caso a condição seja falsa pelo botão “senão” (figuras 8.10 e 8.11).

O autor pode também desejar criar vários comportamentos para uma dada ação. Neste caso ele utiliza a opção múltiplos comportamentos. Os comportamentos serão ativados na mesma ordem em que o autor os definir.

Através do Modo *Script* o autor pode visualizar o que criou no Modo Visual. Na Figura 8.12 é mostrado o *script* do comportamento definido para o objeto porta.

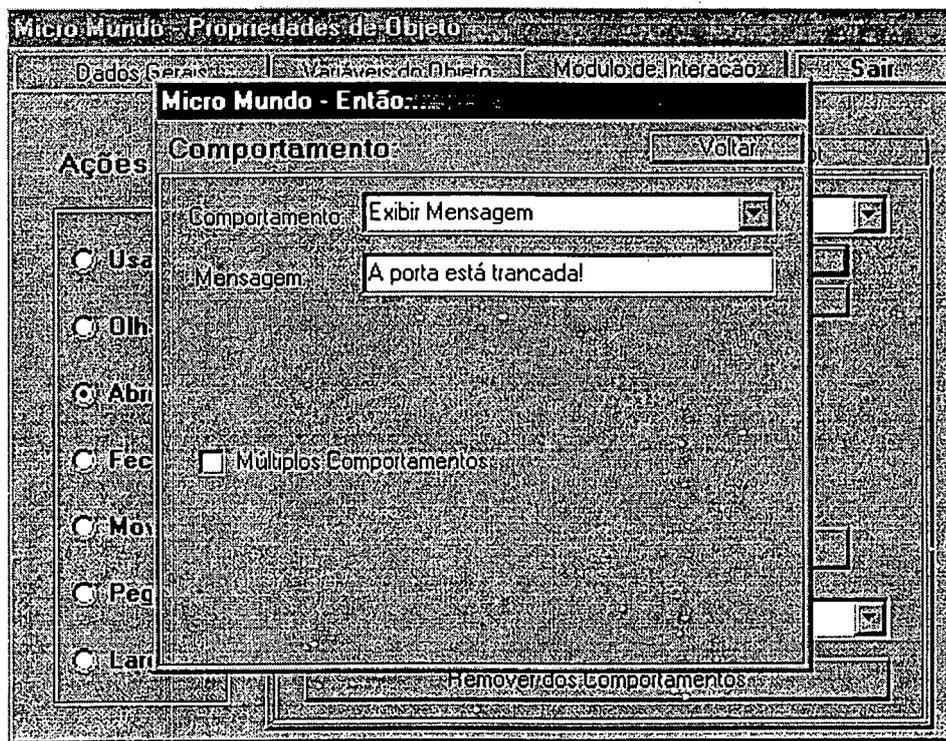


Figura 8.10: Tela de definição para o comportamento do botão “então”

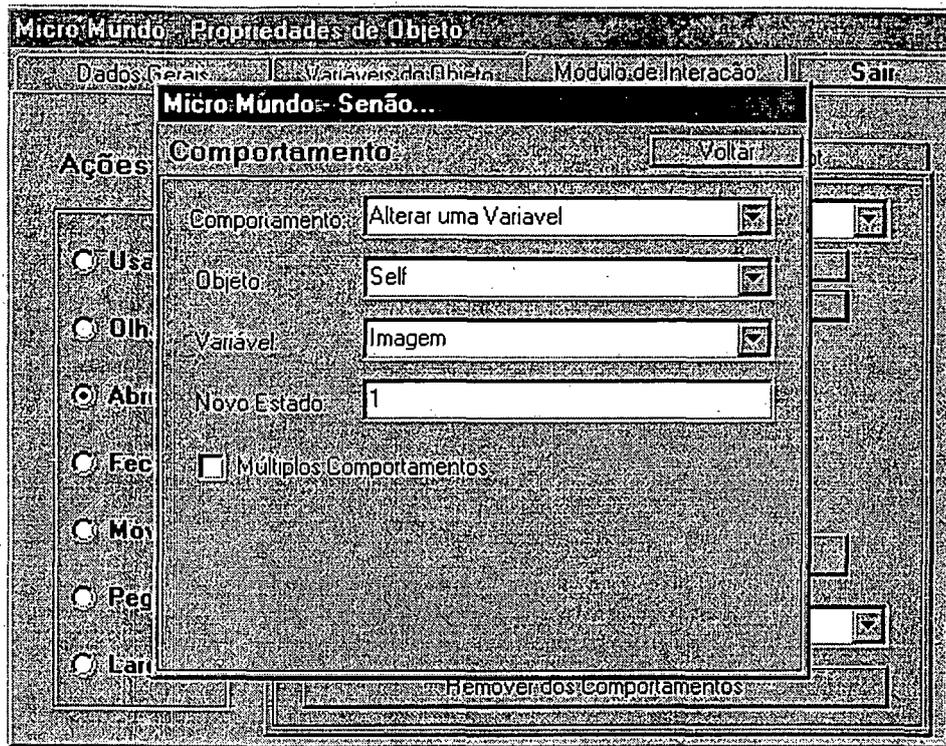


Figura 8.11: Tela de definição para o comportamento do botão “senão”

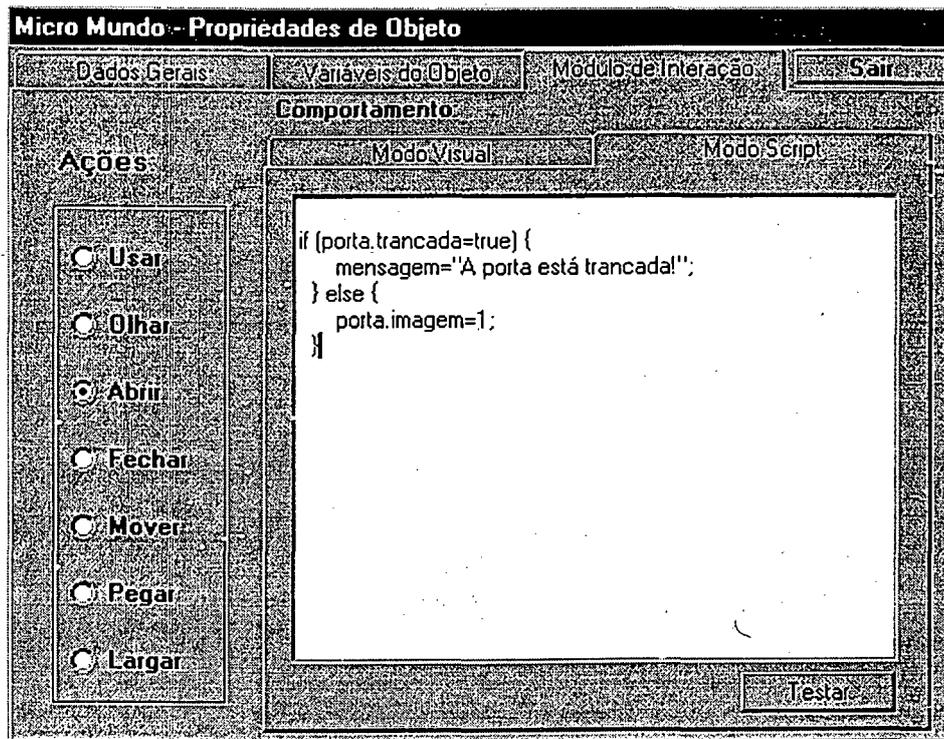


Figura 8.12: Tela do Módulo Interação - Modo Script

Na partição Personagens, através do botão “novo”, o autor escolhe a imagem que deseja associar ao personagem do micro-mundo, conforme mostra a Figura 8.13. Através do botão “propriedades” o autor define se o personagem será um *avatar* ou um *bot*, suas propriedades, forma de interação, etc. Ou seja, este botão aciona o **Módulo de Interação** para personagens, mas não está disponível nesta versão.

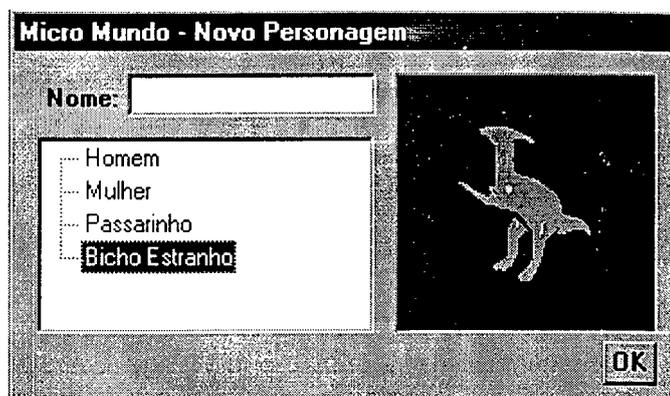


Figura 8.13: Biblioteca de Personagens

A Barra de Ferramentas (Figura 8.14) possui as seguintes funções: copiar, recortar, colar, visualização tridimensional, girar objeto, mais zoom, menos zoom, deslocar objeto e seleção de objeto. A função de cada botão é mostrada ao simples passar do mouse sobre o botão.



Figura 8.14: Barra de Ferramentas do Módulo de Composição

O Campo de Visualização representa o cenário e auxilia o autor na distribuição dos objetos. Algumas operações podem ser feitas no Campo de Visualização, como por exemplo, ao simples passar do mouse sobre o objeto, é apresentado seu nome, e com um duplo-clique ativa-se a janela Propriedades do Objeto. Utilizando a Barra de Ferramentas, o autor pode operar os objetos, movê-los, copiá-los, etc.

9. ÁREA DA APLICAÇÃO

Este capítulo versa sobre a área de projetos estruturais, que faz parte dos currículos dos cursos de Engenharia Civil e Arquitetura. O objetivo deste capítulo é proporcionar uma visão geral acerca do que é tratado dentro desta área.

9.1. Princípios Gerais do Projeto Estrutural

9.1.1. Peças Estruturais

Toda construção, para que possa cumprir suas finalidades, deve possuir um conjunto de partes resistentes. Entendida como um conjunto de partes resistentes da construção, a estrutura deve garantir a existência de uma certa segurança contra estados limites, nos quais a construção deixa de cumprir as suas finalidades [FUS 77].

Em virtude da complexidade das construções, a realização das estruturas requer o emprego de diferentes tipos de peças estruturais, as quais devem ser adequadamente combinadas na formação do conjunto resistente.

O ponto de partida do projeto estrutural de uma construção consiste na idealização de um arranjo estrutural, com o qual se pretende que todas as partes da construção possam ter a sua resistência assegurada. Para a idealização desse arranjo estrutural é necessário conhecer o comportamento de cada uma das partes da estrutura a ser projetada. Para isso torna-se conveniente que as peças estruturais sejam classificadas segundo o seu comportamento.

As estruturas distinguem-se, de acordo com o tipo de capacidade resistente que apresentam, em estruturas lineares e estruturas laminares [LEO 82a]. Vigas e pilares, por exemplo, são estruturas lineares constituídas de barras, enquanto que placas, chapas e cascas fazem parte das estruturas laminares formadas por superfícies - lajes.

O princípio estrutural funciona da seguinte maneira: a laje de cobertura é diretamente apoiada nas vigas, as quais percorrem toda a borda da laje. Nos cantos, ou encontros de duas vigas, iniciam-se os pilares, os quais apoiam-se no solo, ou a laje do piso, sendo esta também “amarrada” por vigas em toda sua borda. Para que toda esta estrutura possa se sustentar no

solo, existe mais uma parte estrutural chamada de fundação. Esta é formada por blocos que seguem a direção dos pilares no interior do solo.

Em poucas palavras, a laje de concreto distribui suas cargas uniformemente nas vigas, as quais por sua vez absorvem esta carga em toda a sua extensão e as distribuem pontualmente nos pilares que transportam as mesmas para as fundações. Veja as figuras 9.1 (a) e (b), a seguir.

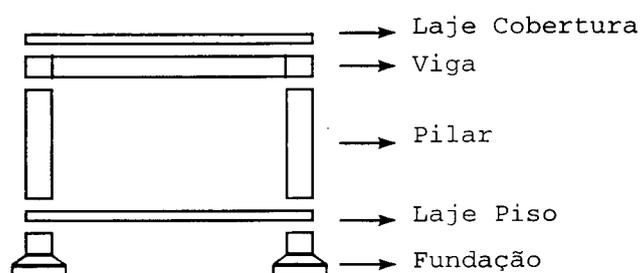


Figura 9.1 (a): Montagem das peças estruturais - imagem corte

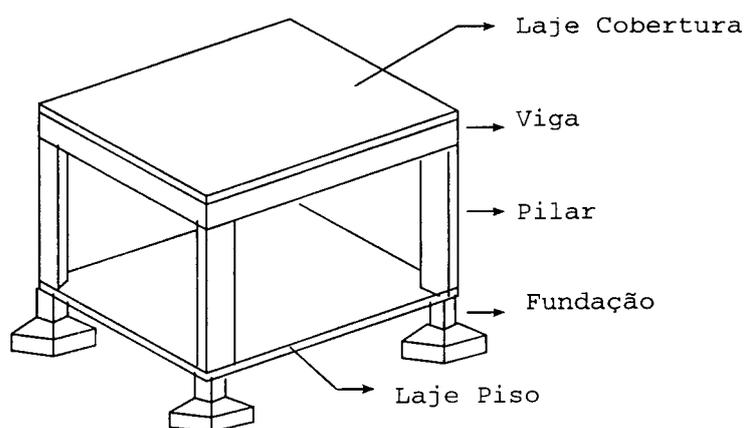


Figura 9.1 (b): Montagem das peças estruturais - imagem isométrica

9.1.2. Análise da Estrutura

A decomposição de uma construção em partes simplifica o problema estrutural, ainda assim são necessárias simplificações ulteriores. De modo geral é pequeno o número dessas partes, daí resultando estruturas excessivamente complexas para um tratamento global [FUS 77].

Para que o projeto estrutural se torne executável, é preciso subdividir de modo virtual a estrutura de cada uma das partes em que foi decomposta a construção. Essa decomposição virtual deve reduzir a estrutura a um conjunto de elementos estruturais suficientemente simples para que cada um deles possa ser tratado globalmente, sem a necessidade de subdivisões ulteriores. Isso se consegue quando cada um dos elementos estruturais considerados pode ser assimilado a um dos modelos estruturais esquemáticos estudados pela Teoria das Estruturas. Esses modelos estruturais correspondem, portanto, a estruturas relativamente simples, cujo cálculo pode ser realizado de modo global. A complexidade desses modelos dependerá dos conhecimentos do projetista e dos recursos disponíveis para o seu cálculo. As simplificações não podem ignorar o comportamento real da estrutura como um todo. A adequação dos modelos estruturais adotados na etapa de análise da estrutura deve ser comprovada ao se realizar a síntese estrutural, etapa indispensável do projeto, mesmo que seja feito de forma simplista.

De modo geral, a maneira mais espontânea de se chegar ao delineamento do arranjo estrutural de uma construção é através da análise das cargas que a solicitam. O delineamento é feito empregando-se o projeto de arranjo geral como sistema de referência. Habitualmente, o projeto de arranjo geral é chamado de projeto arquitetônico, não obstante seja apenas uma parte do mesmo.

Como já foi mencionado, nos casos em que se empregam placas (lajes), estas exigem o emprego simultâneo de barras (vigas, pilares e fundações). A fim de se manter a espessura das placas dentro de limites razoáveis, os comprimentos de seus lados devem ser limitados. Para isso, as placas são subdivididas em painéis por meio de barras de sustentação.

A espessura mínima de lajes de concreto armado para carga acidental distribuída deve ser de 7 cm.; no caso de coberturas sem carga acidental, 5 cm.; no caso de lajes sujeitas a tráfego, 10 cm., quando destinadas a automóveis, e 12 cm. quando destinadas a caminhões.

A esbelteza da laje l/h que depende do vão não deve ser adotada muito pequena, para evitar flechas muito grandes, para evitar fissuras indesejáveis em paredes divisórias sobre pisos.

Nas estruturas de concreto as lajes, quase sempre, são ligadas monoliticamente às vigas que as suportam. Entretanto como se tratam de peças estruturais de duas categorias distintas, com métodos de cálculo próprios, na maioria dos casos é inexecutável a consideração desse monolitismo do ponto de vista prático. Em lugar disso, por meio de uma subdivisão virtual separam-se as lajes das vigas. Essa separação é válida para a consideração exclusiva das lajes, retomando-se a ligação existente, na consideração das vigas.

As vigas e pilares são também peças estruturais fundamentais numa construção. As vigas na sua maioria têm largura com o mesmo dimensionamento da parede. A altura varia em função da carga e do vão que deve suportar. Porém, em casos onde os vãos são muito extensos, ou a carga da laje ou de outros pavimentos superiores forem muito grandes, pode-se variar a sua largura para que a sua altura não ultrapasse as alturas convencionais. O mesmo acontece com os pilares, os quais preferencialmente são adequados às dimensões das paredes num dos seus lados [LEO 82b].

As dimensões das fundações dependem da qualidade do solo. Em geral elas não ultrapassam na profundidade em três vezes a maior dimensão da fundação (em silos e obra semelhantes, duas vezes a largura da obra) e dos recalques que a estrutura pode tolerar. Quando a própria estrutura for rígida na direção vertical, as desigualdades do solo são compensadas por uma redistribuição das cargas nos pilares, essa redistribuição, porém, precisa ser avaliada no que diz respeito aos seus efeitos nos esforços internos da superestrutura.

9.1.3. Fissuras

A baixa resistência à tração do concreto é a razão pela qual as peças estruturais de concreto fissuram-se até com tensões de tração pequenas. Deve-se levar em consideração que na maioria dos casos, além das tensões de tração provenientes de esforços de coação internos ou externos (tensões de coação ou tensões intrínsecas), as condições desfavoráveis de temperatura e umidade do ar podem atingir valores tão elevados, que o concreto se fissura antes mesmo do carregamento atuar. Este perigo de fissuração existe principalmente nas primeiras dez a quarenta horas após a concretagem, enquanto o concreto novo ainda não tem nenhuma resistência a tração e já está sujeito a tensões intrínsecas ou tensões de coação

elevadas devido a variação de temperatura. Tais variações de temperatura surgem devido ao calor de hidratação no interior da peça e ao esfriamento no exterior [LEO 82c].

Outro caso muito comum de fissuras ocorre nas paredes de vedação. Estas surgem devido à falta de exatidão no cálculo dos elementos estruturais de um projeto. Um exemplo é quando as vigas não suportam as cargas das lajes por terem que vencer um vão muito grande entre os pilares. Dessa forma, estas acabam sobrecarregando a parede criando fissuras nas extremidades próximas aos pilares. Existe também o problema do mal cálculo das fundações em relação ao tipo de solo, o mesmo faz com que a estrutura trabalhe em alguns pontos distorcendo e causando fissuras nas paredes.

Generalizando, existem inúmeros casos de fissuras em vedações. Estas na sua maioria ocorrem por falha no cálculo estrutural do projeto arquitetônico, no mal dimensionamento de pilares e vigas, na ausência de pilares em grandes vãos, os quais não podem ser suportados apenas pelas vigas, etc.

9.1.4. Condições gerais

Existem algumas condições básicas nos elementos estruturais para pequenos projetos arquitetônicos e de preferência térreos, que numa análise rápida já permitem verificar as necessidades e às vezes até as dimensões dos elementos estruturais.

Nestes casos, numa visão geral, já se percebe os vãos suportáveis de uma viga e a quantidade de pilares necessários para que a viga não se sobrecarregue. A importância e o papel dos elementos estruturais e destas condições básicas, já devem ser conhecidas pelos alunos dos dois primeiros anos dos cursos de Engenharia Civil e de Arquitetura quando passam a trabalhar diariamente no desenvolvimento dos projetos arquitetônicos.

10. APRESENTAÇÃO DA APLICAÇÃO

Este capítulo apresenta um protótipo de um micro-mundo aplicado à área de engenharia civil relacionada ao projeto estruturado. Tal micro-mundo é composto de uma obra acabada, mas com problemas estruturais.

A aplicação protótipo apresenta problemas no arranjo estrutural decorrentes da falta de exatidão no cálculo dos elementos estruturais do projeto, acarretando assim o surgimento de fissuras nas estruturas de concreto e paredes. No exemplo apresentado são previstos apenas problemas relacionados ao peso excedente sobre vigas e pilares da obra. Outros fatores determinísticos de fissuras, tais como: umidade do ar, temperatura, fundação, entre outros, não foram considerados neste protótipo.

O protótipo foi implementado com a integração das linguagens VRML 2.0 e Java, e pode ser visualizado através do *browser Community Place* da Sony.

A tela de abertura da aplicação exibe o micro-mundo prototipado, composto pelo cenário e pelo menu de ações (usar, abrir, olhar, empurrar, fechar, pegar, puxar e largar) para a manipulação dos objetos, o qual inclui o botão do *container* do usuário. Esta tela é apresentada na Figura 10.1, sobre o *browser* utilizado, o qual possui ícones e botões padrões para navegação dentro do ambiente.

O usuário do ambiente se encontra na situação de um engenheiro civil que precisa detectar e avaliar as causas dos problemas na obra, para então poder solucioná-los. Para isto o usuário tem plena liberdade sobre os objetos contidos no micro-mundo, como por exemplo, vigas, pilares, portas, etc.

O engenheiro já possui de saída uma trena - localizada no *container*, com a qual pode fazer medições nas paredes, vigas e pilares da casa. Para tanto, o usuário deve utilizar o botão “usar” do menu de ações juntamente com a trena sobre o objeto que queira medir.

As vigas e pilares empilhadas no lado de fora são de diferentes tamanhos, e podem ser tomadas pelo engenheiro para realização de testes no interior da casa. Estes testes consistem

na colocação de uma viga ou pilar em determinadas posições e verificação do resultado. O usuário deve utilizar do comando “pegar” junto ao objeto que deseja selecionar. Este objeto é carregado para dentro do *container* do usuário até que este queira utilizá-lo ou deixá-lo em algum outro lugar. Para tanto, utiliza-se o comando “largar” do menu de ações.

O usuário através do botão “olhar” adquire mais informações sobre um determinado objeto, como por exemplo do que este é constituído. E, através dos botões “abrir” e “fechar”, abre e fecha objetos, no caso, portas e janelas.

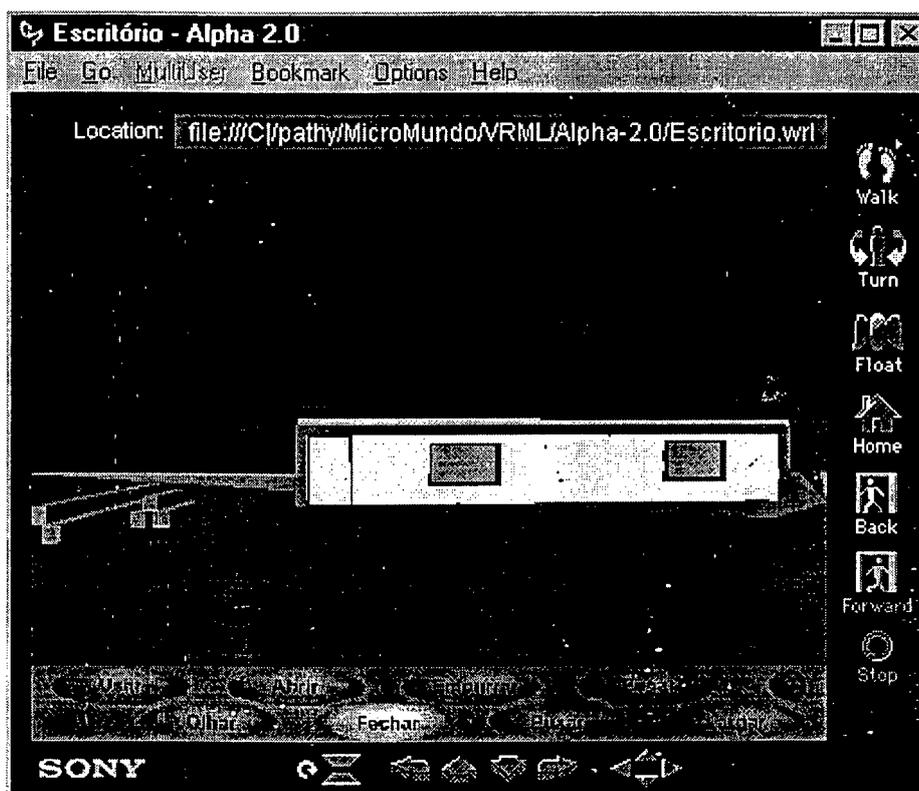


Figura 10.1: Tela de apresentação

Na Figura 10.2, pode-se ter uma visão interna da obra, possibilitando visualizar algumas fissuras, a título de ilustração.

A colocação de uma viga e dois pilares nos locais apropriados permitem fazer com que estas fissuras desapareçam. A colocação das vigas ou pilares em locais inadequados poderão não resolver o problema, ou resolve-lo parcialmente, ou resolve-lo a um alto custo, caso um

número excessivo de vigas sejam usadas, ou ainda comprometendo mais ainda a construção se uma viga for posicionada inadequadamente.

A filosofia deste sistema não é ensinar cálculo estrutural, mas permitir que o estudante que esteja acompanhando esta disciplina tenha um ambiente no qual ele pode desenvolver sua capacidade de percepção dos problemas a serem solucionados, tentar resolvê-los através de experimentos, e com os resultados colhidos, avaliar seu próprio conhecimento sobre o domínio da situação em que se encontra.

O ambiente se caracteriza por ser uma simulação da prática de um engenheiro civil. Além do fato de ser uma forma muito mais atrativa do que o desenho bidimensional dos vegetais e cópias heliográficas, os quais já estão se tornando instrumentos obsoletos.

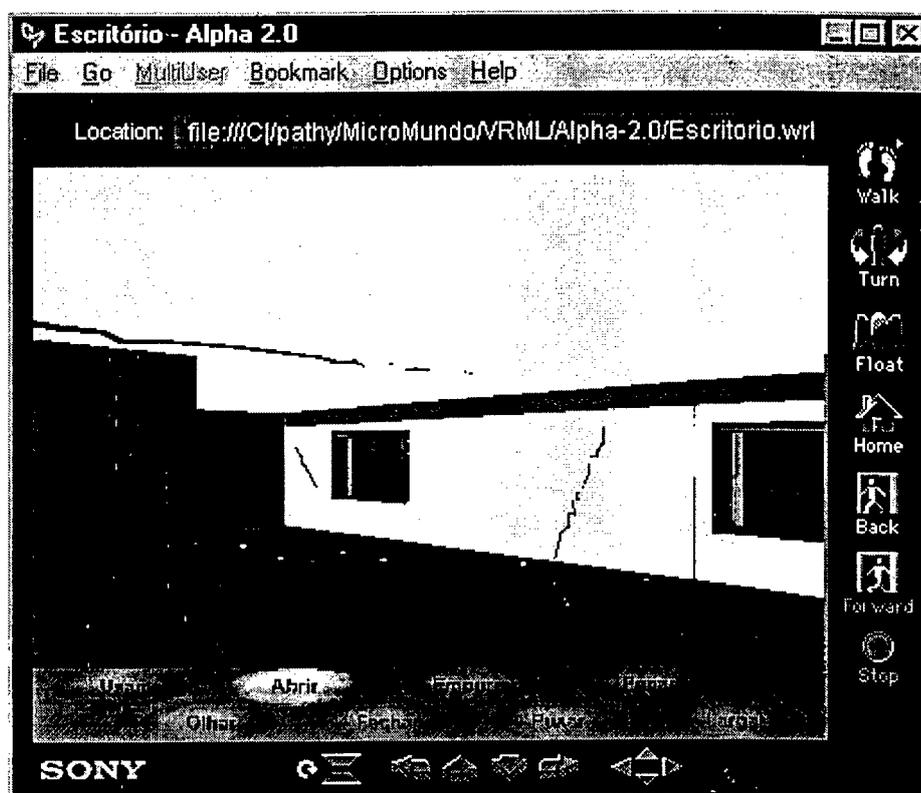


Figura 10.2: Visão interna do micro-mundo

11. CONSIDERAÇÕES FINAIS

Este capítulo apresenta algumas considerações finais acerca da metáfora dos jogos educativos, do protótipo da ferramenta de autoria além do modelo desenvolvido para a aprendizagem de projetos estruturais.

11.1. *Metáfora dos Jogos Educativos*

O paradigma criado pela metáfora dos jogos educativos é o de ter o usuário como um personagem do contexto ao qual está inserido. O usuário é parte integrante do ambiente e interage através do papel que representa dentro deste contexto, deixando de ser um mero espectador. Tais jogos promovem o desenvolvimento de habilidades de pensamento e raciocínio e incorporam motivação ao ambiente, oferecendo assim, de uma maneira natural, meios para desenvolver tais habilidades. Com estes jogos aprende-se também a cooperar, barganhar, respeitar e projetar conseqüências de longo prazo em um cenário.

Sob este paradigma o ambiente deve possuir um contexto altamente exploratório. Dessa forma o usuário vai interagir com o ambiente através de várias atividades relacionadas a este mundo no qual está inserido, fortalecendo e acelerando a construção de suas estruturas cognitivas, ou seja, ele estará aprendendo durante esse processo exploratório.

O ambiente deve ser construído de forma que o aprendiz passe por várias fases apenas quando constrói o conhecimento associado a cada uma delas. É claro que também pode ser acrescido conhecimento na forma textual, mas apenas quando necessário durante a interação.

Neste tipo de ambiente o aprendiz está no controle. Eles promovem os aspectos individuais e a criatividade, uma vez que centram-se na descoberta e na invenção de possibilidades de resolução. O controle do aprendiz também estimula um conjunto de resultados efetivos, dos quais pode-se citar: o aumento dos níveis de atenção e a realização de atividades mais positivas. Contudo o maior resultado efetivo está refletido na autonomia do aprendiz para criar o seu próprio conhecimento.

11.2. Ferramenta de Autoria

Através da construção do primeiro protótipo da ferramenta de autoria, pode-se avaliar com clareza todos os aspectos que envolvem a construção de um ambiente de tal porte. A pesquisa envolveu também um estudo sobre as linguagens VRML e Java, ambas sem um histórico definido, pois estão em contínua adaptação e crescimento. Vários testes foram realizados para garantir a identidade destas linguagens com a proposta deste projeto.

As pesquisas feitas também concluíram a inexistência de uma ferramenta com características semelhantes à desenvolvida. O projeto que mais se assemelha ao desenvolvido é o ExploreNet - descrito no capítulo 6.

O protótipo da ferramenta foi desenvolvido em Delphi 2.0, que era mais conhecida da autora e até então supriu todas as necessidades requeridas para este primeiro protótipo.

Esta implementação serviu como base para se fazer um análise dos aspectos desenvolvidos, tais como a relação entre os módulos, a forma como estes seriam modelados dentro da ferramenta, etc. Uma grande preocupação foi torná-la agradável o suficiente para que leigos em informática se sentissem a vontade em criar micro-mundos com tal ferramenta.

Pode-se constatar também que o uso de comportamentos pré-definidos em uma simples lista de comportamentos desejados não deveria ser um conceito tão desafiador. Entretanto, com este método, o repertório de ações possíveis é as vezes um pouco limitado para prover comportamentos com uma dada liberdade de “exploração”. Em contrapartida notou-se através da criação da aplicação, que muito dessa limitação pode ser banida pela criatividade do autor e de sua familiarização com a ferramenta.

Espera-se na próxima versão prototipada, poder construir uma ferramenta com uma capacidade maior de criar os diversos modos de tratamento com os objetos, além do módulo de interação com os personagens. A experimentação da atual versão deverá apontar ainda uma série de outras necessidades que serão também implementadas.

11.3. Aplicação

O protótipo construído para demonstração do potencial da ferramenta de autoria, mostrou que, o mesmo possui muito mais possibilidades de crescimento do que se havia pensado inicialmente. Estas possibilidades incluem criar situações problemas que relacionem também problemas de hidráulica, elétrica, e outras áreas da engenharia.

Com a criação deste protótipo pode-se comprovar que a ferramenta de autoria vai de encontro às idéias do projeto. É lógico que isto tornar-se-á cada vez mais evidente conforme novas versões prototipadas da ferramenta forem criadas e por conseguinte outras aplicações geradas a partir desta.

O exemplo mostrado permite inferir que com um pouco de criatividade a ferramenta pode se adequar às mais diversas áreas de ensino, tais como: literatura, línguas estrangeiras, física, química, entre outras.

11.4. Linhas de Pesquisa

Pode-se notar que o projeto em questão envolve várias áreas de pesquisa, como redes de computadores (conexão remota, largura de banda, protocolos de comunicação, etc.), inteligência artificial distribuída (teoria de agentes), processamento de linguagem natural e realidade virtual (mundos tridimensionais).

Deve-se fazer uma especificação para a ferramenta de autoria sobre os aspectos relacionados a rede de computadores já mencionados.

É necessário também desenvolver um modelo de sistema multi-agentes que suporte um módulo de processamento de linguagem natural, e seja adequado à ferramenta em questão.

Um aspecto muito importante a ser lembrado em qualquer construção de *software*, são os aspectos ergonômicos, que podem ser analisados através deste primeiro protótipo.

12. REFERÊNCIAS BIBLIOGRÁFIAS

- [AME 96] AMES, Andrea L.; NADEAU, David R. & MORELAND, John L. **The VRML Sourcebook**. John Wiley & Sons, Inc. NY, 1996.
- [BOR 93] BORGES, Karen S.; SILVA, Mauro C.P. & LIMA, Vera L.S. **Uma Ferramenta para o Desenvolvimento de Software Instrucional**. *ANAIS do IV Simpósio Brasileiro de Informática na Educação - SBIE*. Recife, PE, 1993. pp. 65-70.
- [BRE 73] BREARLEY, Molly & HITCHFIELD, Elizabeth. **Guia Prático para Entender Piaget**. IBRASA Editora. São Paulo, SP. 1973.
- [BYR 96] BYRNE, Christine M. **Water on Tap: The Use of Virtual Reality as an Educational Tool**. Dissertação submetida como requerimento parcial para obtenção do grau de Doutor em Filosofia. University of Washington, Department of Industrial Engineering, 1996.
- [CAL 96] CALVERT, Charles. **Delphi 2 - Unleashed**. SAMS Publishing, Indianapolis. 2ª Edição. 1996.
- [CAR 95] CARDOSO, Claudine Maya. **AALO - Ambiente de Aprendizagem de Lógica**. Projeto de graduação apresentado ao curso de Ciências da Computação, UFSC. Florianópolis, SC, 1995.
- [CAS 96] CASAS, Luis A. A., BRIDI, Vera L. & FIALHO, Francisco A. P. **Construção de Conhecimento por Imersão em Ambientes de Realidade Virtual**. *ANAIS do VH Simpósio Brasileiro de Informática na Educação - SBIE*. Belo Horizonte, MG, 1996. pp. 29-43.
- [CLU 95] CLUNIE, G.; CAMPOS, G. & ROCHA, A. **Caracterização de um Meta-ambiente de Aprendizagem baseado em Hipertecnologias**. *ANAIS do VI Simpósio Brasileiro de Informática na Educação - SBIE*. Florianópolis, SC, 1995. pp. 386-397.
- [COU 90] COUTAZ, J. **Interfaces homme-ordinateur**. Dunod. Paris, 1990.

- [CRO 95] CROMBY, John, STANDEN, Penny & BROWN, David. **Using Virtual Environments in Special Education**. VR in the Schools. Vol. 1, N° 3. Dec, 1995.pp. 1-3.
- [DED 94] DEDE, C. & LOFTIN, B. **The design of Artificial Realities to Improve Learning Newtonian Mechanics**. [online] Disponível pela Internet via WWW: <URL: <http://www.vetl.uh.edu/ScienceSpace/newton.html>>.
- [DED 95] DEDE, Chris. **The Evolution of Constructivist Learning Environments: Immersion in Distributed, Virtual Worlds**. [online] Disponível pela Internet via WWW: <URL: <http://www.virtual.gmu.edu/cdede.htm>>.
- [DED 96] DEDE, C., SALSMAM, M. & LOFTIN, B. **ScienceSpace: Virtual Realities for Learning Complex and Abstracts Scientific Concepts**. [online]Disponível pela Internet via WWW:<URL:<http://www.vetl.uh.edu/ScienceSpace/absvir.html>>.
- [DOL 87] DOLLE, Jean Marie. **Para compreender Jean Piaget: uma iniciação a psicologia genética piagetiana**. Editora Guanabara Koogan S.A. 4ª Edição. Rio de Janeiro, RJ. 1987
- [FAQ 96a] **FAQ sobre linguagem JAVA-** [online] Disponível pela Internet via WWW <URL: <http://sunsite.unc.edu/javafaq/javafaq.html>>.
- [FAQ 96b] **FAQ sobre MUDs** - [online] Disponível pela Internet via WWW <URL: <http://www.math.okstate.edu/~jds/mudfaq-p1.html>>.
- [FAQ 96c] **FAQ sobre VRML** - [online] Disponível pela Internet via WWW <URL: http://www.oki.com/vrml/VRML_FAQ.html>.
- [FIA 94] FIALHO, Francisco A. P. **Modelagem computacional da equilibração das estruturas cognitivas como o proposto por Jean Piaget**. Tese de doutoramento apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina. Florianópolis, SC. 1994.
- [FLO 96] FLOHR, Udo. **O Espaço no Ciberespaço**. BYTE - Brasil. Vol.05, N° 03. Março, 1996. pp. 40-43.

- [FOL 92] FOLEY, J. et alii. **Computer Graphics, Principles and Practice**. Addison Wesley. 1992.
- [FUS 77] FUSCO, Péricles B. **Estruturas de Concreto - Fundamentos do Projeto Estrutural**. McGraw-Hill. São Paulo, SP. 1977.
- [GAL 93] GALNANDER, S. & NILSSON-NYSTRÖM, K. **A Aldeia Viking**. Inova. Lisboa, 1993.
- [GUE 96] GUERICKE, K. **What is VRML?** - [online] Disponível pela Internet via WWW <URL: <http://livedv.com/Whitepapers/vrml.html>>.
- [HED 96] HEDBERG, S. **Virtual Reality at Boeing: Pushing the Envelope**. Virtual Reality Special Report, Jan, 1996.
- [HEL 96] HELSEL, Sandra K. **VR in Education**. VR News. Vol. 5 Issue 6. Jul, 1996. pp. 31-34
- [HIR 95] HIROTA, K. & HIROSE, M. **Providing force feedback in Virtual Environment**. IEEE Computer Graphics and Application. Sep, 1995. pp. 22.
- [HOL 93] HOLM, I.; PALMBER, R. & BANG-NIELSEN, E. **Alice: jogo de aventuras baseado na obra "Alice no país das maravilhas"**. Inova. Lisboa, 1993.
- [HOL 95] HOLLANDS, Robim. **Sourceless Trackers**. VR News. Vol. 4, Nº 3. Apr, 1995.
- [ISD 93] ISDALE, Jerry. **What is Virtual Reality?** [online] Disponível pela Internet via WWW <URL: <http://www.cms.dmu.ac.uk/People/cph/VR/whatisvr.html>>.
- [KRU 91] KRUEGER, M. W. **Artificial Reality II**. Addison-Wesley. 1991.
- [LAQ 94] LAQUEY, Tracy & RYER, Jeanne C. **O Manual da INTERNET - Um guia introdutório para acesso às redes globais**. Editora Campus. Rio de Janeiro, RJ, 1994.
- [LEO 82a] LEONHARDT, F. & MÖNNING, E. **Construções de Concreto - Princípios Básicos do Dimensionamento de Estruturas de Concreto Armado**. Editora Interciência. Vol. 1. Rio de Janeiro, RJ. 1982.

- [LEO 82b] LEONHARDT, F. & MÖNNING, E. **Construções de Concreto - Princípios Básicos sobre a Armação de Estruturas de Concreto Armado**. Editora Interciência. Vol. 3. Rio de Janeiro, RJ. 1982.
- [LEO 82c] LEONHARDT, F. & MÖNNING, E. **Construções de Concreto - Verificação da Capacidade de Utilização, Limitação da Fissuração, Deformações, Redistribuição de Momentos e Teoria das Linhas de Ruptura em Estruturas de Concreto Armado**. Editora Interciência. Vol. 4. Rio de Janeiro, RJ. 1982.
- [LER 94] LERNER, M. **Uma Avaliação da Utilização de Jogos na Educação**. *ANAIS do III Encontro da Educação com a Informática: você fazendo acontecer*. Faculdade Carioca, R.J., 1994.pp. 103-105.
- [LIM 80] LIMA, Lauro de O. **Piaget para Principiantes**. Summus. São Paulo, 1980.
- [MOS 95] MOSHELL, J. M., HUGHES, C. E. & ROSENTHAL, J. **Building ExploreNet Worlds - A Prototypical Set of Hands-On Tutorials**. [online] Disponível pela Internet via WWW <URL: <http://longwood.cs.ucf.edu/ExploreNet/tutorials/Xnet.Tutorial.Intro.html>>.
- [NOV 95] **O tira-teima do construtivismo - 50 grandes e pequenas dúvidas esclarecidas**. Nova Escola. Março,1995.pp. 8-13.
- [OSB 95] OSBERG, Kimberley M. **Virtual Reality and Education: Where Imagination and Experience Meet**. VR in the Schools. Vol. 1, Nº 2. Sep, 1995.pp. 1-3.
- [PAN 93] PANTELIDIS, Verônica S. **Virtual Reality in the Classroom**. Educational Technology. Vol. 33. Apr, 1993.pp. 23-27.
- [PAN 95] PANTELIDIS, Verônica S. **Reasons to Use Virtual Reality in Education**. VR in the Schools. Vol. 1, Issue 1. Jun, 1995.pp. 9.
- [PAN 96] PANTELIDIS, Verônica S. **Suggestions on When to Use and When Not to Use Virtual Reality in Education**. VR in the Schools. Vol. 2, Nº 1. Jun, 96.pp. 18.
- [PAV 95] PAVEL, P.; CAMPOS, G.; ROCHA, A. & RABELO, A. **Sisautor: um Sistema de Autoria para Construção de Tutores Hipermídia em Cardiologia**. *ANAIS*

- do VI Simpósio Brasileiro de Informática na Educação - SBIE*. Florianópolis, SC, 1995.pp. 375-385.
- [PES 95] PESCE, Mark. **VRML - Browsing and Building in Cyberspace**. New Riders Publishing, Indianápolis, 1995.
- [PIA 76] PIAGET, J. **A Equilíbrio das Estruturas Cognitivas - Problema Central do Desenvolvimento**. Editora Zahar. Rio de Janeiro, 1976.
- [PIA 87] PIAGET, J. & GARCIA, R. **Psicogênese e história das ciências**. Publicações Dom Quixote. Lisboa, 1987.
- [PIA 91] PIAGET, J. **O nascimento da inteligência na criança**. Editora Guanabara. Rio de Janeiro, 1991.
- [PIN 96a] PINHO, Márcio S. **Realidade Virtual como Ferramenta de Informática na Educação. Apostila do Tutorial**. VII Simpósio Brasileiro de Informática na Educação. Belo Horizonte, MG. 1996.
- [PIN 96b] PINHO, Márcio S. **Uma Experiência do Uso de VRML no ensino de Computação Gráfica em Curso de Graduação em Informática**. ANAIS do VI Simpósio Brasileiro de Informática na Educação - SBIE. Belo Horizonte, MG, 1996.pp. 431-436.
- [RAM 95] RAMOS, Edla M. F. **Análise Ergonômica do Sistema HiperNet buscando o Aprendizado da Cooperação e da Autonomia**. Projeto de Tese submetido ao programa de Pós-graduação em engenharia de produção, UFSC. Florianópolis, SC, 1995.
- [REP 95] **Repositório VRML** - [online] Disponível pela Internet via WWW <URL:<http://www.sdsc.edu/SDSC/Partners/vrml/examples.html>>. San Diego Supercomputer Center, Universidade da Califórnia. San Diego.
- [ROB 84] ROBINETT, Warren. **Manual do Rocky's Boot. The Learning Company**, 1984.
- [ROC 96] ROCHA, Helder. **Comparando linguagens: Java e C++**. LAN TIMES - Brasil. Vol.02, Nº 03. Junho, 1996. pp. 30-39.

- [ROE 96] ROEHL, Bernie. **VRML - Tech Talk**. WEBSmith Magazine. Nº 01. Janeiro, 1996. pp.17-18.
- [RUS 95] RUSSELL, Stuart & NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. Prentice Hall Series in Artificial Intelligence, New Jersey. 1995.
- [STO 96] STOCK, Juanita K. **Virtual Reality Brings Literature to Life at Haywood Community College**. VR in the Schools. Vol. 1. Issue 4. Mar, 1996.pp 8.
- [TAL 96] TALKMITT, Marcia. **VESAMOTEX - Virtual Education Science and Math of Texas**. VR in the Schools. Vol. 1 Issue 4. Mar, 1996.pp. 5-7.
- [ULB 94] ULBRICHT, Vânia R. **Modelagem Construtivista do Módulo do Estudante de um Sistema Auxiliado por Computador para a Geometria Descritiva**. *Proposta de Tese apresentado para exame de qualificação. Pós-graduação em engenharia de produção, UFSC. Florianópolis, SC, 1994.*
- [WAL 84] WALLACE, Mike & GRIMM, Leslie. **Manual do Robot Odyssey**. The Learning Company, 1984.
- [WAZ 93] WAZLAWICK, Raul Sidnei. **Um Modelo Operatório para Construção de Conhecimento**. Tese de Doutorado, EPS-UFSC. Florianópolis, SC, 1993.
- [WIN 93] WINN, William. **A Conceptual Basis for Educational Applications of Virtual Reality**. [online] Disponível pela Internet via WWW <URL: http://www.hitl.washington.edu/projects/learning_center/r-93-9.html>.
- [WOO 95] WOOLDRIDGE, Michael & JENNINGS, Nicholas R. **Intelligent Agents: Theory and Practice**. Springer-Verlag, Berlin.1995.

ANEXO A - LINGUAGENS UTILIZADAS

A1. A LINGUAGEM JAVA

A1.1. Descrição geral

A linguagem Java é uma linguagem orientada a objetos criada no início dos anos noventas pela Sun Microsystems. O projeto original da Sun, chamado Oak, era criar uma linguagem leve com códigos pequenos e que servisse para controlar produtos eletrônicos de consumo. Em meados de 1994, a Sun percebeu que a Word Wide Web teria espaço para essa linguagem, caracterizada pela independência de plataformas e sistemas operacionais e concebida para redes [FAQ 96a].

A linguagem Java possui muitas semelhanças com C, C++ e Objective C (como exemplo, pode-se citar os *loops for*, os quais possuem a mesma sintaxe nas quatro linguagens); mas não é baseada em nenhuma delas, e na verdade, nenhum esforço foi feito para que fosse compatível com elas.

Algumas vezes, Java é chamada de C++ ++ -- . A linguagem foi originalmente criada porque C++ se mostrou inadequada para algumas tarefas. Uma vez que os projetistas não estavam preocupados com compatibilidade com outras linguagens, eles adicionaram algumas coisas que C++ não possui, como *garbage collection* (coleta de lixo) e *multithreading*; bem como descartaram algumas características de C++ que provaram ser melhores na teoria do que na prática como múltipla herança e sobrecarga de operadores.

Uma questão muito relevante a ser observada, é que Java foi desenvolvida com a idéia da execução de código seguro através da rede, mesmo quando a fonte daquele código fosse não-confiável e possivelmente maliciosa. Isto requer a eliminação de algumas características de C e C++. O exemplo mais evidente é a inexistência de ponteiros, de modo que programas Java (ao menos na teoria), não acessem endereços arbitrários na memória.

Além disso, uma outra questão fundamental é o fato de Java ter sido desenvolvida para ser “inter-plataforma”, não somente em código fonte, mas também em sua forma binária

compilada. Como isso é impossível entre diferentes processadores, a forma binária de um programa Java compilado consiste em um código de bytes (*bytecode*) que é interpretado em tempo real pelo interpretador Java. Assim, para portar programas Java para outras plataformas, basta portar o interpretador.

Em última análise, tem-se o fato de que Java foi projetada para possuir menos erros de lógica (*bugs free code*) em seu código. Tipicamente, programas C têm em média um erro (*bug*) a cada cinquenta e cinco linhas de código, geralmente ligados a alocação de memória. Java faz algumas implementações para tornar a ocorrência de *bugs* menos comum:

- a) possui tipagem forte;
- b) a linguagem é pequena e fácil de ler e escrever, de modo a se tornar fluente com facilidade;
- c) não há construções indefinidas ou dependentes de arquitetura;
- d) é orientada a objetos, aumentando a possibilidade de reuso de código;
- e) possui concorrência.

A1.2. Relação Java versus WWW

Em um nível mais alto, Java adiciona várias características a “*sites*” WWW existentes:

- a) Gráficos arbitrários - Java permite ao servidor desenhar em uma janela no cliente, de forma que uma página *Web* se comporte como um programa que desenha em uma janela;
- b) Tipos de dados arbitrários - Java permite que se mostre qualquer tipo de dados particular. Por exemplo, ao invés de se carregar um *bitmap* de uma figura, carrega-se código Java que interprete o formato original da figura, que vem em seu formato original;
- c) Menos carga no servidor - ao invés do servidor executar *scripts* cgi-bin, os programas Java são executados nos clientes, diminuindo a carga do servidor;
- d) Mais interação com o usuário - Java permite também coletar informações do usuário do programa Java, ao invés de simplesmente mostrar coisas na tela.

A1.3. Plataformas

A Sun elaborou um kit de desenvolvimento disponível para os sistemas operacionais: Sparc Solaris, Windows NT em Intel, Windows 95, MacOS 7.5 em PowerMac e Macs 68020, 68030 e 68040.

Várias empresas já portaram Java para outras plataformas como: IRIX, Linux, OS/2, AIX, UnixWare, HP/UX e Digital UNIX. Em andamento estão implementações para NextStep, SunOS 4.1 e Amiga, entre outros.

O Netscape Navigator 2.0 executa *applets* Java em Windows NT, Windows 95, Solaris, SunOS 4.1, Linux e a maioria das outras plataformas UNIX, além da versão Mac.

A1.4. Ferramentas

Já existem algumas ferramentas e ambientes para desenvolvimento da linguagem Java, como ambientes IDEs para desenvolvimento de *applets* Java para compiladores C++, como também o Java Workshop, escrito em Java e lançado pela Sunsoft; o Symantec Café, criado pela Symantec para Symantec C++, Windows 95 e Windows NT; o Roaster, da Natural Intelligence, para PowerMac; e a última versão do C++ da Borland, a 5.0, que possui um *set* de ferramentas para desenvolvedores Java.

A1.5. Pesquisando Java

O assunto Java pode ser encontrado em *sites* WWW, listas de discussões, *newsgroup*, e até mesmo por FTP. Para isto, basta procurar em índices WWW no mundo, como o Yahoo ou Pointcom. Existem também muitos livros sobre o assunto.

A2. A LINGUAGEM VRML

A2.1. Introdução

O WWW é baseado em protocolos de rede, tais como o HTTP, e um número de formatos que descrevem como imagens, som e vídeo são representados frente a diferentes tipos de computadores que são conectados à Internet. O mais importante formato hoje é o

HTML, o qual permite textos e gráficos serem exibidos juntos usando a metáfora de página bidimensional. A última tendência entre *browsers* HTML tais como Netscape Navigator é incluir capacidades de autoria HTML tão bem, que faz com que muitas pessoas que inicialmente estavam contentes somente em usar a web, estão tornando-se interessadas em publicar sua própria *homepage*.

Conforme afirma Guericke, em [GUE 96], pode-se dizer que VRML é para 3D o que HTML é para 2D. Enquanto HTML especifica como documentos 2D são representados, VRML é um formato que descreve como ambientes 3D podem ser explorados e criados na WWW. Desde que 2D é realmente apenas um subconjunto de 3D, qualquer objeto 2D pode ser facilmente representado num ambiente 3D. No livro de Mark Pesce - “*VRML - Browsing and Building in Cyberspace*” [PES 95], Tim Berners-Lee, o “pai” da web, afirma que VRML é o futuro da web por ser mais natural às pessoas estarem imersas em um espaço 3D do que clicarem através de páginas hiperlinkadas.

A2.2. Definição

A Linguagem de modelagem de realidade virtual (VRML - *Virtual Reality Modeling Language*) é uma tecnologia inovadora que foi projetada com o intuito de criar ambientes interativos tridimensionais em tempo real para WWW.

VRML é uma linguagem de descrição de cenário que padroniza como ambientes tridimensionais são representados na Web. Diferente das linguagens de programação como C++, VRML não tem que ser compilado e executado. Ao contrário disto, arquivos VRML são analisados e então exibidos. Visto que isto é um processo muito mais rápido, a criação de arquivos VRML é muito mais simples do que a programação, provendo muitas facilidades, além de permitir mais interatividade.

A2.3. Comparando VRML com HTML

VRML permite uma interação muito mais rica do que HTML. Quando visualizada uma *homepage* 2D, as opções são basicamente limitadas a pular de página em página e olhar imagens permanentes, sob uma perspectiva pré-determinada. Quando visita-se mundos

VRML, entretanto, pode-se livremente escolher a perspectiva desejada para visualizar o mundo. Além disso, pode-se navegar livremente através de ambientes 3D, cujos os conteúdos são apenas limitados pela imaginação de seu projetista.

Enquanto se caminha através de um mundo, pode-se pegar objetos e inspecioná-los de todos os lados, e caso a aplicação VRML usada inclua capacidades de autoria, pode-se até criar e modificar objetos 3D. Portanto, VRML permite expandir *homepages* 2D até *homeworlds* 3D.

Os arquivos VRML são usualmente mais compactos do que arquivos de vídeo e frequentemente até menores do que imagens de alta-resolução. As melhores ferramentas VRML permitem acessar e salvar espaços 3D em blocos. Através de enfileiramento pode-se quebrar cenas VRML grandes em pequenos arquivos que podem ser carregados incrementalmente. Este processo pode até funcionar melhor do que em 2D.

Enquanto um *browser* HTML não tem como saber qual imagem ou qual parte da imagem o usuário está mais interessado, um *browser* VRML “sabe” onde o usuário está e em qual direção está olhando. Desde que se tenha um visão de primeira perspectiva em VRML, precisa-se apenas olhar na direção onde se quer ir e o *browser* VRML dará maior prioridade para carregar os objetos que fazem parte desta visão. Os objetos restantes são carregados em segundo plano ou conforme se mude o ponto de vista.

A maioria dos arquivos VRML são atualmente, surpreendentemente pequenos e podem nem requerer um enfileiramento. Arquivos de imagens, som e vídeo, tendem a ser bem maiores porque armazenam cada parte de informação que aparece na tela, bit por bit. Já que esta abordagem seria muito inadequada em três dimensões, arquivos VRML apenas armazenam coordenadas necessárias de cada objeto e algumas instruções para o *browser* 3D de como conectar os pontos e como exibir as superfícies. Visto que o *browser* 3D tem então que gerar o ambiente 3D destas informações, o real gargalo para ambientes VRML de larga escala é o poder da CPU local e afeta apenas uma extensão menor da largura de banda da conexão Internet.

A2.4. Ferramentas para utilização e construção de mundos VRML

No geral, existem três categorias de ferramentas disponíveis hoje em dia:

- a) *Browsers* VRML;
- b) Pacotes de autoria VRML;
- c) Ferramentas de criação 3D.

As ferramentas preferidas para projetistas de jogos e produtores de multimídia têm sido pacotes de modelação e animação tais como: Softimage, 3D Studio, Strata Studio e Caligari true Space.

Algumas ferramentas de criação 3D, permitem salvar modelos em VRML, outras necessitam que se compre utilitários - em separado - para converter o formato interno para VRML.

Mesmo que alguns *browsers*, como Community Place (Sony) e o WebSpace, necessitem de *browsers* HTML para buscar cenas VRML, outros, onde podemos citar WorldView e WorldSpace, somente precisam se comunicar com *browser* HTML quando eles encontram um documento HTML. A maioria dos *browsers* VRML contam com DDE para se comunicar com *browsers* HTML.

As ferramentas de autoria mais populares são Virtus Walkthrough, WebSpace Author, Home Space Builder e Caligari worldSpace. Enquanto a maioria delas permite adicionar *hyperlinks*, somente WebSpace Author executa sobre máquinas SGI e inclui alguma tecnologia sofisticada de redução de polígonos, mas não possui nenhuma ferramenta de derivação 3D como o Caligari worldSpace, que executa sobre o Windows 3.1 e Windows 95 e está disponível de graça para uso não comercial.

A2.5. Formato de um arquivo em VRML

Um arquivo VRML é uma descrição textual do mundo VRML que foi criado. O arquivo VRML é criado num editor VRML, mas pode também ser gerado através de algum utilitário que traduz outros formatos de arquivos gráficos para VRML. O arquivo VRML

possui extensão `.wrl` e é constituído por quatro componentes principais, a saber: o cabeçalho *VRML*, os nós (ambos obrigatórios), os campos e os comentários, (estes dois últimos, opcionais).

```

VRML header   → #VRML V1.0 ascii
comment       → # Created by Pat Doe 1995
               # Hut
               # Draw hut base
node(shape)   → Cylinder {
               field   →      height 2.0
                       radius 0.75
               }
               # Move the pen up
node(property)→ Translation {
               translation 0.0 2.0 0.0
               }
               # Draw the roof
               Cone {
               BottomRadius 2.5
               }

```

Figura A2.1: Exemplo de arquivo VRML [AME 96]

O cabeçalho (*VRML header*) de um arquivo VRML deve conter exatamente o texto mostrado na Figura A2.1, ele descreve para o *browser* que este arquivo é um arquivo VRML, que é compilado com a versão 1.0* da especificação VRML, e que é um arquivo *ascii*.

Os nós (*nodes*) descrevem as formas e as propriedades (cor, intensidade da luz, posição, etc.) do mundo VRML. Geralmente eles contém: o tipo do nó (requerido), um conjunto de chaves (requerido), e um número de campos (opcional), que definem os atributos (como altura e área) dos nós dentro das chaves.

A ordem dos nós dentro do arquivo VRML determina a ordem em que o mundo VRML é construído. VRML lê os nós em seu arquivo de cima para baixo, e segue as instruções de desenho dos nós nesta ordem.

Os campos (*fields*) são opcionais porque cada campo tem um valor *default* que é usado pelo *browser* VRML, caso não seja especificado. Por exemplo, um cubo VRML *default* tem

* A última versão oficial da linguagem VRML é a 2.0.

uma largura de 2.0 unidades VRML, uma altura de 2.0 unidades, e uma profundidade de 2.0 unidades. A ordem dos campos dentro de um nó não é importante, eles podem ser especificados na ordem em que o projetista desejar.

As unidades em VRML não estão relacionadas a nenhuma unidade de medida do mundo real, tais como polegadas ou centímetros. Elas descrevem um tamanho ou uma distância dentro do contexto do mundo VRML. Pode-se pensar numa unidade como uma polegada, um metro ou até mesmo um ano-luz - qualquer unidade de distância que pareça ser a mais conveniente. Porém, estas unidades devem ser consistentes. Caso se tenha medido o tamanho da forma de um cilindro em centímetros, deve-se também medir as propriedades da forma de um cilindro em centímetros, como por exemplo a posição.

Os comentários (*comments*) permitem adicionar informação extra ao arquivo VRML não afetando a aparência de seu mundo. Isto permite adicionar notas para o arquivo sobre quem o criou, quando foi criado, porque foi criado e o que ele contém. Os comentários em um arquivo VRML possuem a mesma função dos comentários em qualquer outra linguagem - a de auxiliar na especificação do arquivo. Eles começam com o sinal # e terminam no final da linha. Qualquer coisa digitada na linha de comentário é ignorada pelo *browser* VRML.

O arquivo VRML descreve como desenhar um mundo tridimensional. Os nós e seus campos definem quais formas são desenhadas, onde cada uma é desenhada, qual é a cor de cada forma, e assim por diante. Tudo isto é definido numa ordem particular, criando no arquivo uma série de direções para o *browser* VRML que descrevem o quê e como desenhar. O *browser* lê o arquivo VRML e segue as instruções de desenho descritas através dos nós e cria o mundo 3D. A ordem é uma questão crítica em VRML. Para se obter os resultados esperados, deve-se cuidadosamente ordenar as direções de desenho no arquivo VRML, imaginado como se o *browser* VRML usasse uma caneta virtual de desenho para mostrar o mundo descrito no arquivo.

Por exemplo, o arquivo da Figura 12.1, tem o propósito de desenhar um chapéu. Sabendo que VRML possui instruções pré-definidas para desenhar cilindros e cones, utilizou-se do cilindro para fazer o corpo do chapéu e colocou-se o cone no topo. O nó *Cylinder* instrui

o browser para desenhar um cilindro (as paredes do chapéu), o nó *Translation* fala ao browser para mover sua caneta virtual de desenho, e o nó *Cone* diz ao browser para desenhar um cone (o teto do chapéu). O resultado pode ser visualizado através da Figura A2.2.

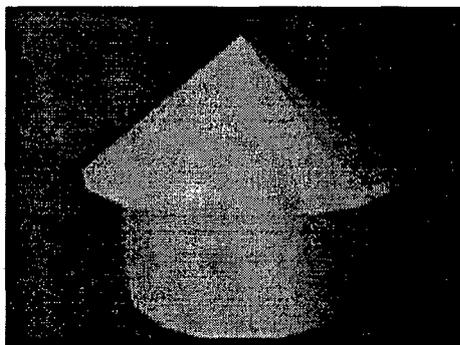


Figura A2.2: Um chapéu

A2.6. Aplicações

Atualmente pode-se encontrar aplicações que utilizam a linguagem VRML nas mais diversas áreas, tais como: arquitetura, artes, astronomia, ciências biomédicas, computação, física, história, matemática, música, química, além de aplicações comerciais e entretenimento [REP 95].

Muito se tem utilizado da tecnologia VRML para se criar *homepages* em três dimensões, dando um aspecto realístico ao cenário. Mas para o navegador que se depara com esta *homeworld* - como assim é chamado as página construídas em VRML, faz-se necessário um *browser* VRML, o qual já pode ser encontrado em diversos *sites* na Internet.