

FUNDAÇÃO DE ENSINO EURÍPIDES SOARES DA ROCHA  
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**CLÉBER GIMENEZ CORRÊA**

**IMPLEMENTAÇÃO E AVALIAÇÃO DE INTERAÇÃO EM UM  
*FRAMEWORK* PARA TREINAMENTO MÉDICO**

Marília – SP  
2008

CLÉBER GIMENEZ CORRÊA

**IMPLEMENTAÇÃO E AVALIAÇÃO DE INTERAÇÃO EM UM  
FRAMEWORK PARA TREINAMENTO MÉDICO**

Dissertação apresentada ao Programa de Mestrado *Stricto Sensu* em Ciência da Computação do Centro Universitário Eurípides de Marília, mantido pela Fundação de Ensino Eurípides Soares da Rocha, para obtenção do título de Mestre em Ciência da Computação (Área de Pesquisa: Realidade Virtual).

Orientadora:  
Profa. Dra. Fátima de Lourdes dos Santos Nunes Marques

Marília – SP  
2008

Dedico este trabalho a toda minha família e a todos os meus amigos, amigos de trabalho, de curso, que me ajudaram muito em todos os momentos, principalmente naqueles mais difíceis.

## **AGRADECIMENTOS**

A Deus, pelo dom da vida e pela ajuda nos momentos difíceis.

À toda minha família, presente em todos os momentos de minha vida, não só durante o curso de mestrado.

À Profa. Dra. Fátima de Lourdes dos Santos Nunes Marques, pela disposição durante a orientação, pela paciência e pelo auxílio para a elaboração de um excelente trabalho.

Aos professores do curso de mestrado, aos revisores, pelo conhecimento e experiências compartilhadas.

À banca examinadora, que contou com os professores Agma e Delamaro, além da professora Fátima, e foram brilhantes na avaliação do trabalho.

Aos amigos do curso de mestrado, companheiros de laboratório, pelo incentivo, pela ajuda nas disciplinas, pelos momentos de descontração. Amigos como: Ana Cláudia, Adriano, Ricardo, Silvio, Wagner, Giu, Juliana, Thaís, Thaís Burity, André, Camila, Sérgio, Reinaldo, Marcos, Bruno, Chico, Rodrigo, Silvia, José Ivo, Richard, Alex Bariri e Alexs Garbato, Bruno, Fábio, Leia, Priscila.

Aos amigos, principalmente aqueles da Prefeitura Municipal de Ribeirão do Sul, que permitiram que eu estudasse, realizando mudanças de horário, mesmo sem entender direito o

que significava Realidade Virtual. Se esquecer de alguma pessoa, que ela me perdoe: Deborat, Maria Manso, Pedro Viana, Fernanda, Douglas, Ivanilda, Rodrigo, Paulo, José Carlos, Tarciso, Deijair, Márcio, Danilo, Rafaela, Madalena, Néia, Juscelino, Janine.

Aos estudantes e professores da Faculdade de Medicina de Marília, pela contribuição na avaliação do projeto.

Ao CNPq, pelo apoio financeiro.

*Não existe verdadeira inteligência sem bondade.*

***Ludwig van Beethoven***

*Somente uma vida consagrada ao demais é digna de ser vivida.*

***Albert Einstein***

*Ensinar é lembrar os outros que eles sabem tanto quanto você.*

***Richard Bach***

*Se queremos progredir, não devemos repetir a história, mas fazer uma história nova.*

***Mahatma Gandhi***

Corrêa, Cléber Gimenez. **Implementação e Avaliação de Interação em um *Framework* para Treinamento Médico**. 185 f. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília. Fundação de Ensino Eurípides Soares da Rocha, Marília, 2008.

## RESUMO

A interação em Ambientes Virtuais envolve navegação, seleção e manipulação de objetos, e controle do sistema, como também a utilização de dispositivos convencionais e não convencionais. Um *framework* com suporte a diversos dispositivos (*mouse*, teclado, luva, equipamento háptico) pode facilitar a construção de aplicações, de acordo com a disponibilidade financeira e grau de realismo desejado. Este trabalho tem como objetivo principal a implementação de um módulo de interação incorporado no *framework* ViMeT (*Virtual Medical Training*), cuja finalidade é prover funcionalidades básicas de Realidade Virtual para construir aplicações de simulação de exames de biópsia. Após a implementação, foi realizada uma avaliação da interação nesses ambientes, obtendo-se ajustes nos requisitos e definição de parâmetros importantes para a construção e aperfeiçoamento de sistemas de Realidade Virtual dentro do domínio citado.

**Palavras-chave:** *Framework*, Treinamento Médico, Interação, Ambientes Virtuais, dispositivos convencionais, dispositivos não convencionais.

Corrêa, Cléber Gimenez. **Implementação e Avaliação de Interação em um *Framework* para Treinamento Médico**. 185 f. Dissertação (Mestrado em Ciência da Computação) – Centro Universitário Eurípides de Marília. Fundação de Ensino Eurípides Soares da Rocha, Marília, 2008.

## **ABSTRACT**

The interaction in Virtual Environments involves navigation, objects selection and manipulation, and system control, as well as the use of conventional and non-conventional devices. A framework with support to several devices (mouse, keyboard, dataglove, haptic equipment) can facilitate the construction of applications, according to financial availability and desired degree of realism. This work has as principal objective the implementation of an interaction module incorporated in ViMeT (Virtual Medical Training) framework, whose purpose is to provide Virtual Reality basic functionalities to building biopsy exams simulation applications. After implementation, it was realized an evaluation of the interaction in these environments, obtaining adjusts in the requisites and definition of important parameters for construction and improvement of Virtual Reality systems considering the cited domain.

**Keywords:** Framework, Medical Training, Interaction, Virtual Environments, Conventional Devices, Non-conventional Devices.



## LISTA DE ABREVIATURAS E SIGLAS

€ Euro

°C Graus *Celsius*

**2D** Bidimensional

**3D** Tridimensional

**5DT** *Fifth Dimension Technologies*

**API** *Application Programming Interface*

**APIs** *Application Programming Interfaces*

**AV** Ambiente Virtual

**AVs** Ambientes Virtuais

**CAVE** *Cave Automatic Virtual Environment*

**CD** *Compact Disc*

**cm** Centímetros

**DOF** *Degree of Freedom*

**Famema** Faculdade de Medicina de Marília

**FOV** *Field of View*

**GB** Gigabytes

**GHOST SDK** *General Haptics Open Software ToolKit – Software Development Kit*

**GHZ** Gigahertz

**GIMP** *GNU Image Manipulation Program*

**GNU** *General Public License*

**GUI** *Graphical User Interface*

**IDE** *Integrated Development Environment*

**IEEE** *Institute of Electrical and Electronics Engineers*

**HDAPI** *Haptic Device API*

**HLAPI** *Haptic Library API*

**HF** *High Force*

**HMD** *Head Mounted Display*

**HMDs** *Head Mounted Displays*

**HOMER** *Hand-centered Object Manipulation Extending Ray-casting*

**HPV** *Human Papiloma Virus*

**HSV** *Herpes Vírus*

**Hz** *Hertz*

**JDK** *Java Development Kit*

**JNI** *Java Native Interface*

**JVM** *Java Virtual Machine*

**Kg** *Quilograma*

**LApIS** *Laboratório de Aplicações de Informática na Saúde*

**LCD** *Liquid Crystal Display*

**LSI** *Laboratório de Sistemas Integráveis*

**MG** *Megabytes*

**mm** *Milímetros*

**MRI** *Magnetic Resonance Imaging*

**ms** *Milissegundos*

**N** *Newton*

**N/mm** *Newton por milímetro*

**PDD** *PHANTOM Device Drivers*

**OpenGL** *Open Graphics Library*

**PHANTOM** *Personal Haptic Interface Mechanism*

**PC** *Personal Computer*

**PC NT** *Personal Computer New Technology*

**PUCRS** *Pontifícia Universidade Católica do Rio Grande do Sul*

**RAM** *Random Access Memory*

**RV** *Realidade Virtual*

**SGI** *Silicon Graphics, Inc.*

**SITEG** *Sistema Interativo de Treinamento em Exame Ginecológico*

**SVGA** *Super Video Graphics Array*

**US\$** *Dólar Americano*

**USP** *Universidade de São Paulo*

**ViMeT** *Virtual Medical Training*

**VHB** *Virtual Haptic Back*

**VRD** *Virtual Retinal Display*

**VRML** *Virtual Reality Modeling Language*

**XP** *eXPerience*

**W** *Weber*

## LISTA DE FIGURAS

Figura 1 - 5DT <i>Ultra Wireless Kit</i> .....	36
Figura 2 – Modelo de <i>Joystick</i> .....	37
Figura 3 - PHANTOM <i>Premium 1.5/6DOF</i> .....	39
Figura 4 – Tesoura háptica com dois graus de liberdade .....	40
Figura 5 – Óculos estereoscópicos .....	41
Figura 6 - 5DT <i>Head Mounted Display</i> .....	42
Figura 7 - Monitor <i>ZScreen 2000i</i> .....	43
Figura 8 - 5DT <i>Binoculars series</i> .....	43
Figura 9 - Monitores estereográficos.....	44
Figura 10 – Exemplo do <i>Workbench</i> Háptico Visual.....	45
Figura 11 - CAVE existente na USP .....	46
Figura 12 - Sala de Controle da CAVE .....	46
Figura 13 - Projeção nas paredes.....	46
Figura 14 - Projetor <i>Zscreen</i> .....	47
Figura 15 - Pesquisador testando simulador de extração de medula óssea .....	50
Figura 16 - VHB em operação.....	51
Figura 17 - Endoscópio .....	53
Figura 18 - Desenho de parte do simulador endoscópico.....	55

Figura 19 - Simulador de Acupuntura Chinesa em execução .....	56
Figura 20 - Dispositivo Háptico <i>CathSim</i> .....	66
Figura 21a – Modelo de <i>CathSim</i> .....	67
Figura 21b – Treinamento com o modelo .....	67
Figura 22 - Deformação virtual da pele durante procedimento.....	68
Figura 23 – Diagrama com classes do ViMeT .....	84
Figura 24 – Projeto da arquitetura do ViMeT .....	85
Figura 25 – Exemplo de aplicação gerada com o ViMeT .....	86
Figura 26 – Exemplo de aplicação gerada sem estereoscopia.....	86
Figura 27 – Exemplo de aplicação gerada com outro objeto virtual deformável.....	87
Figura 28 - 5DT <i>Data Glove 5 Ultra</i> .....	88
Figura 29 - PHANTOM <i>Omni</i> .....	89
Figura 30 – Estrutura do Grafo de Cena em Java3D.....	91
Figura 31 - Diagrama de camadas do <i>OpenHaptics Toolkit</i> .....	95
Figura 32 – Diagrama de um Programa HDAPI .....	96
Figura 33 – Diagrama de um Programa HLAPI.....	97
Figura 34 – Funcionamento do módulo de interação .....	98
Figura 35 – Diagrama de classes atualizado do ViMeT.....	99
Figura 36 – Trecho do código de navegação.....	100
Figura 37 – Diagrama de integração e interação para a luva de dados .....	103
Figura 38 – Diagrama de integração e interação para o dispositivo háptico.....	104

Figura 39 – Código exemplo de uma classe .....	105
Figura 40 – Exemplo de arquivo de cabeçalho .....	106
Figura 41 – Diagrama Hierárquico de tipos nativos.....	107
Figura 42 – Exemplo de código nativo.....	108
Figura 43 – Utilização do teclado.....	109
Figura 44 – Utilização do <i>mouse</i> .....	109
Figura 45 – Diagrama de execução para o dispositivo háptico .....	110
Figura 46 – Botões do PHANTOM <i>Omni</i> .....	112
Figura 47 – Esquema de integração do dispositivo háptico .....	114
Figura 48 – Diagrama de execução para o dispositivo luva de dados.....	116
Figura 49 – Esquema de integração da luva de dados.....	118
Figura 50 – Estrutura de modelagem da mão .....	119
Figura 51 – Inclinação dos dedos virtuais .....	120
Figura 52 – Hierarquia de objetos na construção da seringa virtual .....	120
Figura 53 – Grafo de cena inicial .....	122
Figura 54a – Grafo de cena atualizado do ViMeT .....	123
Figura 54b – Continuação do grafo de cena atualizado .....	124
Figura 55 – Novo AV gerado por meio do ViMeT .....	125
Figura 56 – Trecho de código para incorporação do módulo de interação .....	126
Figura 57 – Tela para seleção de dispositivos na <i>ViMeTWizard</i> .....	127
Figura 58 – Trecho de código da classe <i>Behavior</i> .....	130

Figura 59 – Trecho de código para coletar informações .....	130
Figura 60 – Exemplo de utilização do dispositivo teclado.....	134
Figura 61 – Exemplo de utilização do dispositivo <i>mouse</i> .....	135
Figura 62 – Exemplos de utilização do dispositivo háptico .....	135
Figura 63 – Demonstração de movimentos captados pela luva de dados .....	136
Figura 64 – Deformação dos objetos com cor.....	138
Figura 65 – Deformação em aramado .....	139
Figura 66 – Trecho de código da translação do dispositivo háptico .....	139
Figura 67 – Aplicação gerada anteriormente à construção do módulo .....	140
Figura 68 – Modelagem do tórax com os dutos internos .....	140
Figura 69 – Modelagem tridimensional do tórax .....	141
Figura 70 – Tórax e mão.....	141
Figura 71 – Gráficos representando os dispositivos e o nível de facilidade de uso .....	145
Figura 72 – Gráfico dos dispositivos associados ao instrumento médico .....	145
Figura 73 – Gráfico dos dispositivos associados à mão virtual.....	146
Figura 74 – Resultados do aspecto visual.....	147
Figura 75 – Resultados com os dispositivos convencionais.....	148
Figura 76 – Resultados com os dispositivos não convencionais .....	149
Figura 77 – Janela <i>Project Settings</i> .....	172
Figura 78 – Janela <i>Project Settings</i> , categoria <i>Preprocessor</i> .....	173
Figura 79 – Inserção de bibliotecas .....	174

Figura 80 – Configuração do compilador para a luva de dados .....	175
Figura 81 – Código da <i>Callback</i> principal .....	177
Figura 82 – Código de verificação da <i>Callback</i> .....	178
Figura 83 – Código em C++ para inicialização da luva de dados .....	179
Figura 84 – Código nativo para receber informações da luva de dados.....	180
Figura 85a – Importação de objetos .....	182
Figura 85b – Definição da hierarquia.....	183
Figura 86 – Trecho de código da composição da seringa.....	184



## LISTA DE TABELAS

Tabela 1: Classificação de interação em AVs .....	33
Tabela 2: Classificação quanto às formas de interação .....	34
Tabela 3: Metáforas de navegação .....	34
Tabela 4: Técnicas de seleção .....	35
Tabela 5: Preço de luvas de dados.....	38
Tabela 6: Preço de dispositivos hápticos.....	41
Tabela 7: Preço de outros dispositivos não convencionais.....	44
Tabela 8: Dispositivos não convencionais e operações.....	88
Tabela 9: Especificações do Dispositivo PHANTOM <i>Omni</i> .....	89
Tabela 10: Principais métodos da classe <i>Keyboard</i> .....	100
Tabela 11: Principais métodos da classe <i>NativeGlove</i> .....	101
Tabela 12: Principais métodos da classe <i>NativeHaptic</i> .....	101
Tabela 13: Tipos em Java e JNI.....	107
Tabela 14: Funções do <i>OpenHaptics Toolkit</i> .....	112
Tabela 15: Tempos de execução da tarefa em milissegundos .....	143
Tabela 16: Desempenho da aplicação para os dispositivos em um PC.....	150
Tabela 17: Desempenho da aplicação para os dispositivos em um <i>Notebook</i> .....	150
Tabela 18: Desempenho da aplicação para as combinações em um PC .....	151
Tabela 19: Desempenho da aplicação para as combinações em um <i>Notebook</i> .....	151

## SUMÁRIO

INTRODUÇÃO.....	19
Objetivos.....	21
Justificativa.....	21
Disposição do Trabalho.....	22
CAPÍTULO 1 - INTERAÇÃO EM AMBIENTES VIRTUAIS .....	24
1.1 Técnicas de Interação .....	25
1.1.1 Navegação.....	27
1.1.2 Seleção/Manipulação .....	29
1.1.3 Controle do Sistema.....	32
1.2 Dispositivos Convencionais e Não Convencionais .....	35
1.2.1 Dispositivos Convencionais.....	35
1.2.2 Dispositivos Não Convencionais .....	36
1.3 Considerações Finais .....	47
CAPÍTULO 2 - APLICAÇÕES DE REALIDADE VIRTUAL NA MEDICINA.....	48
2.1 Simulação de Coleta de Medula Óssea .....	49
2.2 Projeto VHB .....	50
2.3 Simulador de Realidade Virtual para Endoscopia.....	53
2.4 Simulador para Treinamento de Acupuntura Chinesa.....	55
2.5 Sistema Interativo de Treinamento em Exame Ginecológico .....	58
2.6 Considerações Finais .....	60
CAPÍTULO 3 - AVALIAÇÃO DE INTERAÇÃO EM AVs .....	62
3.1 Fatores Humanos de Interação.....	64
3.2 Avaliação de Dispositivos Hápticos em Simulações Cirúrgicas .....	65
3.3 Avaliação de Técnicas de Seleção e Manipulação em Mundos Virtuais .....	70
3.4 Uma Avaliação Sobre o Uso de Estímulos Táteis em um AV .....	76
3.5 Considerações Finais .....	80
CAPÍTULO 4 – IMPLEMENTAÇÃO E AVALIAÇÃO DE INTERAÇÃO NO VIMET. ....	81
4.1 ViMeT – <i>Virtual Medical Training</i> .....	83
4.2 Dispositivos do Módulo de Interação .....	87
4.3 Considerações Sobre a Tecnologia Adotada .....	89
4.4 Bibliotecas dos Dispositivos Não Convencionais .....	93
4.5 Visão Geral do Módulo de Interação.....	98
4.5.1 Definição da Interação e Integração com os Dispositivos Não Convencionais .....	102
4.6 Integração entre as Linguagens de Programação Java e C++ .....	104
4.7 Integração dos Dispositivos Convencionais.....	108
4.8 Integração do Dispositivo Háptico .....	109

4.9 Integração da Luva de Dados .....	115
4.10 Construção dos Objetos Tridimensionais .....	119
4.11 Integração com o ViMeT.....	125
4.12 Avaliação de Interação em Aplicações Geradas a Partir do ViMeT .....	127
4.12.1 Metodologia de Avaliação.....	128
4.13 Considerações Finais .....	131
<b>CAPÍTULO 5 – RESULTADOS E DISCUSSÕES .....</b>	<b>132</b>
5.1 Resultados da Implementação .....	133
5.2 Avaliação de Profissionais da Área Médica .....	137
5.3 Considerações Finais .....	152
<b>CAPÍTULO 6 – CONCLUSÕES .....</b>	<b>153</b>
6.1 Contribuição para a Área de Computação.....	153
6.2 Contribuição para a Área de Médica .....	154
6.3 Trabalhos Futuros .....	156
6.4 Limitações .....	157
6.5 Trabalhos Publicados.....	157
<b>REFERÊNCIAS .....</b>	<b>158</b>
<b>APÊNDICE A .....</b>	<b>169</b>
A.1 Questionário Pré-Teste .....	169
A.2 Questionário Pós-Teste.....	170
<b>APÊNDICE B.....</b>	<b>172</b>
B.1 Informações Técnicas de Configuração do Compilador.....	172
<b>APÊNDICE C.....</b>	<b>176</b>
C.1 <i>Callback</i> Principal.....	176
<b>APÊNDICE D .....</b>	<b>179</b>
D.1 Código para Inicialização da Luva de Dados .....	179
D.2 Funções Nativas para Obtenção de Informações.....	180
<b>APÊNDICE E.....</b>	<b>182</b>
E.1 Importação e Definição de Características da Mão Virtual .....	182
E.2 Importação e Definição de Características da Seringa Virtual .....	183

## INTRODUÇÃO

O desenvolvimento de computadores e dispositivos eletrônicos permitiu o surgimento de uma área denominada Realidade Virtual, ou simplesmente RV, a qual pode ser referenciada como a mais avançada forma de interface homem-máquina já criada, consistindo na elaboração de mundos virtuais ou artificiais com o uso de alta tecnologia, para convencer o usuário de que ele se encontra em uma outra realidade (MONTERO, 2003).

Existem outras definições para o termo Realidade Virtual, como: “uma forma de usuários de computadores poderem visualizar, manipular e interagir com dados complexos” (AUKSTAKALNIS, BLATNER, 1992 **apud** KIRNER, 2006); e “uma técnica que permite a criação de ambientes sintéticos, gerados por computador, e a utilização de canais multi-sensoriais, oferecendo ao usuário a possibilidade de realizar operações como navegação, interação e imersão” (KRUEGER, 1991; JACOBSON, 1991; BURDEA, COIFFET, 1994; **apud** KIRNER, 2006). Segundo Burdea e Coiffet (1994), em termos de funcionalidades, RV é uma simulação gerada por computador, usada para criar um ambiente que se pareça com o ambiente real, o qual responde a entradas dos usuários em tempo real, como gesto ou voz, propiciando, desta forma, uma interação.

Um sistema de RV deve possuir alguns requisitos, tais como: interface de alta qualidade, para se aproximar ao máximo do mundo real e permitir uma interação mais intuitiva; alta interatividade, permitindo ao ambiente reagir de forma adequada de acordo com as ações dos usuários; imersão, consistindo na sensação oferecida a um participante de uma simulação de estar dentro do mundo virtual; envolvimento e uso da intuição, que consiste em oferecer condições para que o usuário se concentre e realize atividades como no mundo real; analogia e ampliação do mundo real, permitindo que o Ambiente Virtual seja definido como o

mundo real, com acréscimo de aspectos que não são encontrados neste último (MACHADO *et al.*, 2002d).

As aplicações de RV são empregadas em diversas áreas do conhecimento, tais como: mecânica, treinamento militar, entretenimento, medicina, entre outras, uma vez que é capaz de proporcionar certas vantagens que seriam difíceis de serem alcançadas no mundo real, como por exemplo, estudo de simulações físicas ou reações químicas em tempo real e uma viagem por dentro do corpo humano (SILVA, 2004).

Dentro da área médica, existem diversas linhas de estudo de RV, como: educação médica, planejamento e simulação de cirurgias, simulação de procedimentos como endoscopia virtual, reabilitação e avaliação neuropsicológica (RIVA, 2003). Em ambientes gerados por computador pode haver uma interação entre ser humano e máquina que possibilite navegar, selecionar e manipular objetos dentro do mundo virtual como se o usuário estivesse no mundo real. Esses ambientes criados com técnicas de RV podem ser de grande auxílio no treinamento médico (MACHADO, 2003a).

A interação pode ser efetuada por meio de diversos dispositivos. Existem os dispositivos chamados de “convencionais”, como a tela de vídeo, *mouse*, teclado, e aqueles denominados “não convencionais”, como luvas, capacetes e dispositivos hápticos. Para atingir um grau de realismo mais elevado, um *feedback* tátil propiciado por dispositivos hápticos pode ser uma solução interessante, principalmente em simuladores cirúrgicos (BATTEAU *et al.*, 2004). A sensação tátil é uma forma de percepção dos seres humanos, gerando determinadas informações como, indícios que facilitem a orientação do médico durante uma endoscopia, onde a visão é limitada (KORNNER e MANNER, 2002). Outro recurso interessante utilizado na criação de aplicações de RV é o uso de *frameworks*, que segundo Bastos (2005), oferecem abstração de dispositivos e de sistemas de projeção, suporte a

sistemas distribuídos, grafos de cenas especializados, heurísticas de interação com o Ambiente Virtual e renderização distribuídas.

Um *framework* pode ser definido como um projeto e implementação abstratos para desenvolvimento de aplicações em um determinado domínio de problemas, com a vantagem de considerável reutilização de componentes (BOSCH e MATTSSON, 2007). Devido a esta vantagem, um *framework* orientado a objetos permite uma elevada produtividade quando se trata da construção de *software* (MATTSSON, 1996).

## **Objetivos**

A partir do contexto apresentado, este trabalho tem como objetivo a implementação de um módulo para tratamento de interações com diversos dispositivos, tanto convencionais (teclado e *mouse*), quanto não convencionais (luva de dados e equipamento háptico), em um *framework* em desenvolvimento, denominado ViMeT.

Também constitui objetivo desta dissertação, a avaliação do módulo implementado, levando em consideração aspectos relacionados primeiro à máquina (tempo de resposta, tipo de dispositivo e computador empregados, número de quadros por segundo); e aspectos relacionados ao usuário (grau de percepção, familiarização com dispositivos, facilidade de uso, desempenho na execução de tarefas de simulação).

## **Justificativa**

Um *framework* como o ViMeT permite uma produtividade maior na construção de novas aplicações voltadas ao treinamento de estudantes de Medicina e médicos. A integração de diversas formas de interação e a utilização de diferentes dispositivos de interação entre o

usuário e a aplicação, de acordo com a necessidade dos usuários e recursos disponíveis, pode facilitar o aprendizado com os benefícios oferecidos pela RV. Além disso, uma avaliação de aspectos computacionais e humanos relacionados à interação neste *framework* é útil para o aperfeiçoamento dos módulos de interação e para a definição de parâmetros que podem ser utilizados na avaliação de outras aplicações.

## **Disposição do Trabalho**

Esta dissertação possui, além desta introdução, seis capítulos, a saber:

**Capítulo 1:** apresenta conceitos relacionados à interação em Ambientes Virtuais, às formas e técnicas de interação, bem como a classificação destas em categorias. Disponibiliza também conceitos sobre dispositivos convencionais e não convencionais;

**Capítulo 2:** descreve algumas aplicações da RV na área médica, principalmente projetos que empregam dispositivos não convencionais, como equipamentos que propiciam *feedback* tátil;

**Capítulo 3:** trata da avaliação de interação em Ambientes Virtuais sob a óptica computacional e humana, com a definição de parâmetros mensuráveis, além de discutir tipos de dispositivos empregados na interação;

**Capítulo 4:** traz a metodologia utilizada na implementação do projeto, descrevendo as linguagens e os recursos de programação adotados, as etapas de desenvolvimento, bem como o processo de avaliação do projeto, com os parâmetros de avaliação adotados;

**Capítulo 5:** expõe os resultados alcançados na implementação, e as informações obtidas na avaliação do módulo de interação;

**Capítulo 6:** apresenta as conclusões sobre o trabalho, bem como as limitações e os trabalhos que podem ser realizados futuramente.

Finalizando, estão as referências bibliográficas que serviram como embasamento teórico para a elaboração desta dissertação, os questionários utilizados na avaliação do projeto, e um tutorial de configuração do compilador *Microsoft Visual C++ 6.0* para utilização dos dispositivos não convencionais, apresentados como Apêndice A e Apêndice B, respectivamente.



## CAPÍTULO 1 - INTERAÇÃO EM AMBIENTES VIRTUAIS

Os Ambientes Virtuais (AVs) vêm sendo considerados como a forma mais natural de interação entre homem e máquina, pois permitem que o ser humano use seus sentidos, como tato, audição e visão, de forma semelhante ao mundo real, para realizar operações, enviando e recebendo informações do computador (MINE, 1995).

Segundo Pinho (2000), um Ambiente Virtual (AV) é um cenário dinâmico, gerado com auxílio de computação gráfica e armazenado em um computador, cujas cenas são exibidas em tempo real. Em geral, os AVs proporcionam ao usuário uma sensação de imersão e a possibilidade de manipulação. São construídos pela combinação de tecnologias, que criam uma interação em tempo real com modelos tridimensionais (BISHOP, FCHS; 1992 **apud** BERALDIN *et al.*, 1996). Os AVs podem ser classificados em “imersivos”, nos quais são empregados dispositivos como capacetes, luvas de dados, salas de projeção, e “não imersivos”, que utilizam monitores, teclado, *mouse* comum (GARCIA *et al.*, 2003).

Um dos principais objetivos na construção de um AV imersivo é fazer com que o usuário se sinta dentro do ambiente tridimensional (CAMARERO *et al.*, 2000), entretanto, a um custo maior se comparado com os sistemas não imersivos (FERRO e PAIVA, 2003). Um AV pode representar um ambiente imaginário ou um ambiente real. A interação é uma característica importante, uma vez que se trata de um cenário dinâmico, que é alterado em tempo real, conforme as entradas do usuário (PINHO e REBELO, 2004).

A interação nesse tipo de ambiente ocorre quando o usuário realiza operações de seleção, manipulação e navegação no mundo sintético tridimensional, por meio das denominadas técnicas de interação. Essas técnicas consistem em comandos e dados definidos

na implementação do sistema de RV, enviados a este para a execução de determinadas tarefas (FREITAS *et al.*, 2003).

Para que haja interação, a troca de informações entre usuário e sistema é essencial. Por isso, os dispositivos de entrada e saída devem ser levados em consideração na construção de AVs, bem como as vantagens e limitações dos equipamentos (BOWMAN *et al.*, 2001b). De acordo com Jenkins (1949) **apud** Pinho (2000), existem basicamente três componentes em uma interface homem-máquina:

- **dispositivo de entrada**, para capturar alguma ação ou estímulo emitido pelo usuário, como o seu próprio movimento, por exemplo;

- **função de transferência**, com o objetivo de mapear a ação capturada para um elemento ou elementos controlados pelo sistema, transferindo, por exemplo, o movimento do usuário para um avatar<sup>1</sup>;

- **dispositivo de saída**, com a finalidade de exibir a consequência da ação inicial, como a modificação da posição do avatar na tela do computador.

Como uma técnica de interação constitui-se de comandos acionados pelo usuário, que afetarão o AV, pode-se dizer que ela atua como uma função de transferência (PINHO, 2000).

## 1.1 Técnicas de Interação

Como mencionado anteriormente, as técnicas de interação são responsáveis pela transferência ou mapeamento das entradas do usuário, como posições, movimentos de partes do seu corpo ou comandos, em ações dentro do ambiente tridimensional, gerando uma resposta do sistema, a qual é emitida pelos dispositivos de saída (PINHO, 2000).

---

<sup>1</sup> Representação do usuário no mundo virtual.

As técnicas de interação podem ser classificadas em três categorias (BOWMAN *et al.*, 2001b):

- **navegação**: movimento do usuário dentro do AV durante uma simulação;

- **seleção/manipulação**: a escolha de algo (seleção), um objeto virtual por exemplo, e a modificação de suas características (manipulação), como: tamanho, forma, cor, posição, orientação;

- **controle do sistema**: comandos específicos disponíveis para alterar o estado do sistema.

Outro importante conceito no que diz respeito à interação é o recurso da metáfora. De acordo com Rocha (2004), a metáfora reutiliza situações semelhantes em diferentes contextos para propiciar formas de operação do sistema conhecidas ou reconhecidas pelos usuários. Conforme Pinho (2000), a metáfora facilita o aprendizado, uma vez que tem como princípio um domínio previamente conhecido por parte do usuário. Exemplos de metáforas serão tratados nas seções seguintes.

Quanto às formas de interação, observa-se que diversas técnicas podem ser implementadas, classificadas em três grandes categorias (MINE, 1995):

- **interação direta do usuário**: uma ação do usuário (movimento de uma parte do corpo), resulta em uma ação no mundo virtual, destacando alguns parâmetros, como: reconhecimento de gesto, apontamento, direção do olhar;

- **controles físicos**: envolve a utilização de botões, *joysticks* e outros dispositivos, com o intuito de que o usuário interaja com o AV. Nesta categoria, o tipo de dispositivo afeta a interação a ser executada, já que um tipo de capacete, por exemplo, pode dificultar a movimentação do usuário durante uma simulação;

- **controles virtuais:** um objeto do AV pode ser implementado como um controle a ser utilizado, proporcionando uma flexibilidade e, em contrapartida, podendo gerar certa dificuldade de interação com tal objeto, pois além de servir de controle, ele pode ser selecionado e manipulado.

Outra classificação que pode ser destacada é apresentada por Pinho (2000), que agrupa as metáforas de interação em:

- **diretas:** o usuário atua sobre o objeto virtual ou sobre o seu ponto de vista, alterando a projeção e a perspectiva de um determinado objeto, ou o ângulo e a posição durante o processo de navegação, além da realização da seleção de um ou mais objetos dentre diversos;

- **indiretas:** empregadas geralmente em ambientes “não imersivos”, quando o usuário trabalha sobre uma “entidade intermediária”, um dispositivo especial ou um objeto de interface, denominado de *widget*, como um botão ou um menu, por exemplo. O *widget* pode apresentar, por meio de um menu, os objetos disponíveis no AV para seleção; pode apresentar parâmetros dos objetos a serem manipulados, como valores de orientação, translação, cor, escala desses objetos; e parâmetros do observador durante a navegação, como direção e ângulo de observação.

### 1.1.1 Navegação

A navegação é a forma mais comum de interação e consiste no movimento do usuário dentro do ambiente sintético. É dividida em três categorias (BOWMAN *et al.*, 2001b):

- **exploração**, na qual o participante de uma simulação realiza uma investigação do ambiente;

- **busca**, que consiste na movimentação do participante para um local específico;

- **manobras**, que são caracterizadas por movimentos de alta precisão.

Dois outros parâmetros que se referem à navegação são a direção e a velocidade. Controles físicos e virtuais, ou partes do corpo do usuário, podem determinar para que direção o usuário deve se mover e com qual velocidade, a qual também pode ser determinada por uma constante definida de acordo com o tamanho do AV (MINE, 1995).

A navegação também pode ser classificada considerando dois componentes denominados *travel* e *wayfinding*.

O conceito de *travel* está relacionado ao movimento do ponto de vista do usuário de um lugar para outro. Existem cinco metáforas comuns para técnicas de interação do tipo *travel* (MINE, 1995):

- **movimento físico**: o movimento do corpo do usuário é usado para movê-lo no ambiente;

- **manipulação manual do ponto de vista**: os movimentos das mãos do usuário são usados para definir o movimento deste dentro do AV;

- **apontamento**: consiste na especificação contínua da direção do movimento;

- **navegação baseada em objetivo**: o usuário especifica o destino, e sua representação no AV salta imediatamente para o novo local no ambiente, como uma espécie de “teletransporte”;

- **planejamento de rota**: o usuário define o caminho que percorrerá no AV.

O *wayfinding* é entendido como um processo dinâmico do uso da habilidade espacial e percepção do ser humano quando em um determinado ambiente, com o objetivo de encontrar um local neste, ou seja, o ser humano dispõe do conhecimento que já possui e adquire conhecimento durante a navegação para encontrar o caminho (SATALICH, 2006).

No que diz respeito ao *wayfinding*, algumas técnicas podem ser implementadas, como (PINHO *et al.*, 2007):

- **mapa:** para indicar os lugares e características do AV;
- **placas:** para indicar ao usuário o objetivo, bem como outros locais dentro do AV;
- **bússola:** com o intuito de apontar para o objetivo, auxiliando o usuário na navegação;
- **rastro:** uma representação, como uma seqüência de setas, é colocada nos lugares pelos quais o usuário passou.

Uma preocupação na navegação é a desorientação do usuário durante a interação. Uma forma de resolver tal problema é a utilização de diversas visões e escalas no AV, auxiliando o usuário na percepção espacial (FUKATSU *et al.*, 1998). Geralmente, os AVs oferecem aos participantes, um ponto de vista e uma escala 1:1 (CONWAY *et al.*, 2006). O uso de um mundo em miniatura similar ao mundo virtual, com um ícone no mundo em escala menor representando o usuário, sendo atualizados simultaneamente durante a interação, auxilia na percepção. Entretanto, pode confundir o usuário, uma vez que este possui dois pontos de vista (BROCKWAY *et al.*, 2006).

### 1.1.2 Seleção/Manipulação

Para manipular um determinado objeto, é preciso primeiramente selecioná-lo. Seleção é a indicação de qual objeto ou objetos o usuário manifesta desejo de interagir no AV (FORSBERG *et al.*, 2006). Desta forma, é necessária a utilização de mecanismos ou técnicas para identificar um ou mais objetos que se deseja manipular, podendo destacar-se (MINE, 1995):

- **entrada de voz:** cada objeto possui um nome ou um identificador, que deve ser conhecido pelo usuário, para que ele possa identificá-lo em um grupo e selecioná-lo ao pronunciar seu nome;

- **seleção de lista:** uma lista com os nomes ou identificadores dos objetos é apresentada e o usuário pode selecioná-los utilizando a voz, por exemplo.

Um problema existente é a necessidade de um *feedback* para o usuário, com o intuito de confirmar o objeto selecionado, indicando que a operação de seleção realmente ocorreu. Isso pode ser feito por meio de recursos visuais ou auditivos, entre outros, como por exemplo, com o uso de um *bounding box*<sup>2</sup> (MINE, 1995).

A seleção também pode ser dividida em (HUFF *et al.*, 2006): **local**, na qual o objeto a ser selecionado está ao alcance do usuário e; **à distância**, na qual o objeto em questão não está ao alcance do participante da simulação, não havendo, portanto, um contato direto. Nesse caso, há necessidade do uso de técnicas, como apontamento, para indicar o objeto, na qual um feixe de luz é lançado da mão do usuário em direção ao objeto virtual.

Algumas técnicas de interação para seleção foram desenvolvidas, tais como (BOWMAN *et al.*, 2001b):

- **Go-Go:** consiste de um mapeamento não linear da extensão da mão do usuário aplicado na representação da mão do mesmo, propiciando a extensão desta representação até que o objeto alvo seja alcançado;

- **Ray-casting:** trata-se da utilização de um raio, partindo por exemplo, da representação da mão do usuário, sendo utilizado para apontar um objeto no ambiente;

---

<sup>2</sup> Consiste em uma “caixa” que envolve um determinado objeto no Ambiente Virtual.

- **Spotlight:** faz uso de um cone de luz, que é lançado em uma determinada direção, sendo que os objetos dentro deste cone podem ser selecionados;

- **Mundo em miniatura:** objetos são selecionados em uma representação em menor escala do AV, podendo inclusive ser manipulados na representação, o que é refletido no AV em escala normal.

Alguns estudos foram realizados com o intuito de desenvolver variantes dessas técnicas, bem como combinações das mesmas, criando, desta forma, novas técnicas de interação, como a HOMER (*Hand-centered Object Manipulation Extending Ray-casting*), que consiste na combinação das técnicas *ray-casting* e mão virtual (BOWMAN, HODGES, 1997).

Uma vez que o objeto ou objetos estão selecionados, a operação de manipulação pode ser realizada, a qual consiste na alteração do posicionamento e rotação do objeto (BOWMAN *et al.*, 2001b). Segundo Hsu (2007), o ser humano pode fazer uso de suas mãos para explorar e manipular objetos no mundo real, sendo esta uma forma natural de interação com o AV. Por isso, equipamentos como luvas de dados são desejáveis, pois permitem que o movimento do usuário (no caso, de suas mãos), seja capturado e enviado por meio de sinais elétricos ao computador, que processa tais sinais e executa alterações no ambiente sintético. Também seria importante que o usuário pudesse visualizar uma representação de sua mão ou mãos dentro do AV.

Para manipular um objeto, modificar sua posição e rotação, controles físicos, como *joysticks*, e controles virtuais, como menus, também podem ser empregados. Um outro ponto a ser definido é o local do centro de rotação do objeto selecionado, que geralmente é aquele localizado na mão do usuário, entretanto, ele pode ser remoto, fazendo com que o usuário faça manipulações em diferentes perspectivas (MINE, 1995).



### 1.1.3 Controle do Sistema

A última categoria de interação, o controle de sistema, consiste em comandos para a comunicação com a aplicação, possibilitando alterações no estado do sistema ou no modo de interação (BOWMAN *et al.*, 2001b; FLASAR, 2000). As técnicas dessa categoria podem ser úteis para facilitar o processo de interação, transferindo, por exemplo, um objeto distante para próximo do usuário por meio de um comando de voz. Neste contexto, se um participante de uma simulação deseja manipular um objeto que está distante de sua área de alcance e precisa acionar uma determinada técnica de navegação para se aproximar deste objeto, o modo de interação necessita ser alterado, finalizando o processamento do evento atual (manipulação de objetos) para chamar uma função de navegação (FLASAR, 2000).

As técnicas de controle de sistema podem ser classificadas em quatro grupos (BOWMAN *et al.*, 2001b; FLASAR, 2000):

- **Sistema baseado em GUI (*Graphical User Interface*):** consiste em comandos representados visualmente, como menus gráficos, os quais podem ser implementados no AV em uma, duas ou três dimensões. A implementação de menus envolve uma série de questões, tais como: localização dos menus no ambiente, forma de acesso, dispositivos empregados, qualidade da visualização das opções (MINE, 1995);

- **Comando de voz:** o sistema interpreta comandos de fala do usuário por meio de um reconhecedor de voz;

- **Interação via gesto:** comandos são representados por movimentos do corpo ou partes do corpo do usuário, como o movimento dos dedos e das mãos;

- **Ferramenta:** consistem em controles físicos e virtuais, como pedais e rodas de determinados dispositivos, e objetos dentro do próprio mundo virtual, que representem comandos para o sistema.

As técnicas dos quatro grupos apresentados podem ser combinadas, pois dependendo da situação durante o processo de interação, uma técnica pode ser mais útil do que outra (FLASAR, 2000). As tabelas a seguir apresentam um resumo das categorias de técnicas de interação, classificações destas, as metáforas de interação, bem como as formas de interação.

Na Tabela 1 pode-se observar as categorias que agrupam as técnicas de interação (navegação, seleção/manipulação e controle do sistema), bem como suas diversas classificações. Nota-se na Tabela 2 que as formas de interação (direta, por controles físicos e por controles virtuais), podem estar presentes em implementações de técnicas nas três categorias de interação.

Tabela 1: Classificação de interação em AVs

<b>Categoria</b>	<b>Classificação</b>
Navegação	<i>Travel</i>
	<i>Wayfinding</i>
	Busca
	Exploração
	Manobra
Seleção	Local
	À distância
	Voz
	Lista
Manipulação	Escala (x, y, z)
	Translação (x, y, z)
	Rotação (x, y, z)
Controle do Sistema	Gesto
	Voz
	Baseado em GUI
	Ferramenta

Tabela 2: Classificação quanto às formas de interação

<b>Formas de Interação</b>	<b>Categorias</b>
Direta	Navegação
	Seleção
	Manipulação
	Controle do Sistema
Controles Físicos	Navegação
	Seleção
	Manipulação
	Controle do Sistema
Controles Virtuais	Navegação
	Seleção
	Manipulação
	Controle do Sistema

A Tabela 3 apresenta uma classificação da navegação, que pode ser dividida em *travel* e *wayfinding*, e suas respectivas metáforas de interação.

Tabela 3: Metáforas de navegação

<b>Tipo</b>	<b>Metáforas</b>
<i>Travel</i>	Movimento físico
	Baseada em objetivo
	Apontamento
	Manipulação manual do ponto de vista
	Planejamento de rota
<i>Wayfinding</i>	Mapa
	Placa
	Bússola
	Rastro

Com relação à seleção, a Tabela 4 apresenta algumas técnicas de interação, sendo que a última delas é resultado da combinação de duas técnicas.

Tabela 4: Técnicas de seleção

<b>Categoria</b>	<b>Técnicas de Interação</b>
Seleção	Go-Go
	<i>Ray-casting</i>
	<i>Spotlight</i>
	Mundo em miniatura
	HOMER

## 1.2 Dispositivos Convencionais e Não Convencionais

Como descrito anteriormente, os outros componentes que tornam possível a interação homem-máquina, além da função de transferência, que engloba as técnicas de interação, são os dispositivos de entrada e saída. Estes permitem ao usuário estabelecer uma comunicação com o sistema, enviando e recebendo dados do computador. Nesta seção são apresentadas algumas tecnologias de dispositivos, dando ênfase às luvas e aos equipamentos hápticos, que estão diretamente relacionados com a proposta deste projeto.

### 1.2.1 Dispositivos Convencionais

Em AVs, a interação pode ocorrer por meio de dispositivos comuns, como *mouse*, teclado, *joystick*, tela do computador (PINHO, 2000). Segundo Hsu (2007), estes são mais familiares aos usuários, entretanto, são limitados quando há a necessidade de executar determinadas tarefas. Como exemplo, salienta-se que a maioria dos *mouses* possui dois graus de liberdade, permitindo uma movimentação somente em duas direções.

É possível a interação em ambientes tridimensionais não imersivos com tais dispositivos, com o usuário realizando operações de seleção, manipulação, navegação, visto

que por meio de comandos do teclado ou do *mouse*, pode-se escolher um objeto, alterar sua posição, orientação e escala, alterar o ângulo de visão e a direção do observador. Por exemplo, com o teclado o usuário aciona *sliders*<sup>3</sup> ou insere valores para definir propriedades de um objeto (como posição, orientação e escala) e com o movimento do *mouse* e seus botões, o usuário pode selecionar objetos, realizar rotação, translação ou alteração no tamanho dos mesmos (PINHO, 2000).

### 1.2.2 Dispositivos Não Convencionais

Para uma interação mais próxima da realidade, aos sistemas de RV podem ser acoplados dispositivos denominados não convencionais, como luvas de dados, capacetes, óculos estereoscópicos, equipamentos hápticos, sensores corporais, entre outros.

O reconhecimento de gesto é uma forma de interação eficiente e altamente intuitiva para AVs (EISENSTEIN *et al.*, 2003). Por isso, as luvas de dados são desejáveis em sistemas de RV. A Figura 1 mostra um par de luvas desenvolvido pela empresa *Fifth Dimension Technologies*.



Figura 1- 5DT Ultra Wireless Kit (*Fifth Dimension Technologies*, 2007)

---

<sup>3</sup> Um componente de interface que permite ao usuário escolher valores dentre uma determinada faixa de valores.

As luvas, geralmente constituídas de fibra sintética e sensores ópticos, procuram captar os movimentos das mãos, identificando o ângulo dos dedos e a posição e a orientação do pulso, por meio dos sensores (THALMANN, 2007). De uma maneira geral, conseguem capturar a postura e a localização das mãos com acurácia e alta velocidade (KOIKE *et al.*, 2001).

Na Figura 2, pode-se observar um *joystick*, dispositivo utilizado amplamente em jogos, e em AVs, geralmente para a navegação, consistindo em um bastão a ser manipulado pelo usuário (indicando direção, ângulo), e botões para controle do jogo ou sistema.



Figura 2 – Modelo de *Joystick*

Em projetos de luvas, geralmente são utilizados sensores mecânicos ou de fibra óptica, com o intuito de captar os movimentos dos dedos. No caso da fibra óptica, um fio ou cabo óptico com junções é empregado no dispositivo para transferência de informações para o computador. Por isso, quando o usuário move seus dedos, conseqüentemente move as juntas, ocasionando a dobra do fio e a redução de passagem de luz, o que indica um movimento. Existem diversos tipos de luvas de dados no mercado, com variação de preço, número de sensores, desempenho, acurácia (MACHADO *et al.*, 2002b). A Tabela 5 apresenta algumas luvas de dados existentes no mercado, bem como seus preços e fabricantes.

Tabela 5: Preço de luvas de dados (EST - *Engineering Systems Technologies*, 2007 e 5DT - *Fifth Dimension Technologies*, 2007)

<b>Nome do Produto</b>	<b>Fabricante</b>	<b>Preço</b>
5DT <i>Glove 5 Ultra</i>	<i>Fifth Dimension Technologies</i>	US\$ 995
5DT <i>Data Glove 14 Ultra</i>	<i>Fifth Dimension Technologies</i>	US\$ 5,495
5DT <i>Data Glove MRI Series</i>	<i>Fifth Dimension Technologies</i>	US\$ 3,495
5DT <i>Data Glove 16 MRI</i>	<i>Fifth Dimension Technologies</i>	US\$ 6,995
5DT <i>Data Glove Ultra Wireless Kit</i>	<i>Fifth Dimension Technologies</i>	US\$ 1,495
<i>CyberGlove System (18 sensores)</i>	<i>Immersion</i>	€ 13.632
<i>CyberGlove2 System (18 sensores)</i>	<i>Immersion</i>	€ 11.177
<i>CyberGlove System (22 sensores)</i>	<i>Immersion</i>	€ 19.995
<i>CyberGlove2 System (22 sensores)</i>	<i>Immersion</i>	€ 16.177
<i>CyberTouch System</i>	<i>Immersion</i>	€ 21.995
<i>CyberGrasp System</i>	<i>Immersion</i>	€ 59.995
<i>CyberForce System</i>	<i>Immersion</i>	€ 109.995

Outros dispositivos ditos como não convencionais usados em ambientes sintéticos são os dispositivos hápticos. O termo háptico pode ser definido como a ciência do tato, envolvendo força e sensação propiciada pelo toque (BURNS *et al.*, 2004). Com o tato, o ser humano pode perceber quando há um contato entre a pele e um objeto (BREWSTER *et al.*, 2000).

O toque consiste de uma sensação quando a pele é submetida a estímulos mecânicos, elétricos, térmicos, químicos (CHOLEWIAK, COLLINS, 1991 **apud** BURDEA, 1996). Esta sensação pode ser dividida em senso háptico e controle senso-motor. No que diz respeito ao senso háptico, leva-se em consideração o tato, que consiste de um conjunto de eventos que se iniciam com estímulos sobre a pele, tais como pressão ou vibração, os quais são captados por receptores (termoreceptores, receptores mecânicos, entre outros), onde descargas elétricas são geradas e transmitidas ao cérebro pelos nervos que, por sua vez, registra a sensação. O controle senso-motor está ligado ao fato de que os seres humanos combinam o senso de

posição e cinestesia, ou seja, postura e movimentos, para exercer um controle motor durante uma atividade (BURDEA, 1996).

Segundo BURDEA (1996), quando se discute a geração de força e a sensação tátil, alguns conceitos devem ser levados em conta, como:

- **feedback tátil:** sensação aplicada à pele que indica alguma sensação;
- **feedback de força:** retorno de sensação de peso ou resistência de algo;
- **feedback cinestésico:** percepção de movimentos por órgãos existentes em músculos, tendões, juntas;
- **feedback proprioceptivo:** movimentos definidos por informações oferecidas de acordo com a postura (juntas do esqueleto).

Os dispositivos hápticos são desejáveis em aplicações de RV, uma vez que a interação homem-máquina em AVs é promovida por meio de interfaces intuitivas (KIM *et al.*, 2002). A Figura 3 exhibe o equipamento háptico PHANTOM *Premium* 1.5/6DOF, fabricado pela *SensAble Technologies*, que oferece 6 graus de liberdade.



Figura 3 - PHANTOM *Premium* 1.5/6DOF (*SensAble Technologies*, 2007)

Com um equipamento háptico, o usuário tem a possibilidade de utilizar o senso de toque para enviar e receber informações do computador, pois o dispositivo é movimentado



pelo usuário e pode produzir um retorno de força na superfície da sua pele (BREWSTER *et al.*, 2000). Como exemplo de informação sobre o ambiente fornecida ao usuário durante a interação, tem-se o sentimento de textura e peso de objetos, podendo haver ou não uma integração com estímulos sonoros e visuais (KOPPER *et al.*, 2006).

Os equipamentos hápticos podem ser projetados como instrumentos cirúrgicos (endoscópios, agulhas, por exemplo), proporcionando aos usuários um grau mais elevado de realismo. Na Figura 4 é apresentada uma tesoura háptica com dois graus de liberdade, sendo um de rotação em um determinado eixo, com resolução angular de 0,056 graus e um retorno de força de no máximo 7,58 N; e o outro de translação em um determinado eixo, com resolução de 0,0152 mm e um retorno de força de 159 N (JAFRY *et al.*, 2003). Em simuladores de coleta de medula óssea, de acordo com a camada da pele, forças diferentes podem ser transmitidas ao usuário (LOPES *et al.*, 2001).

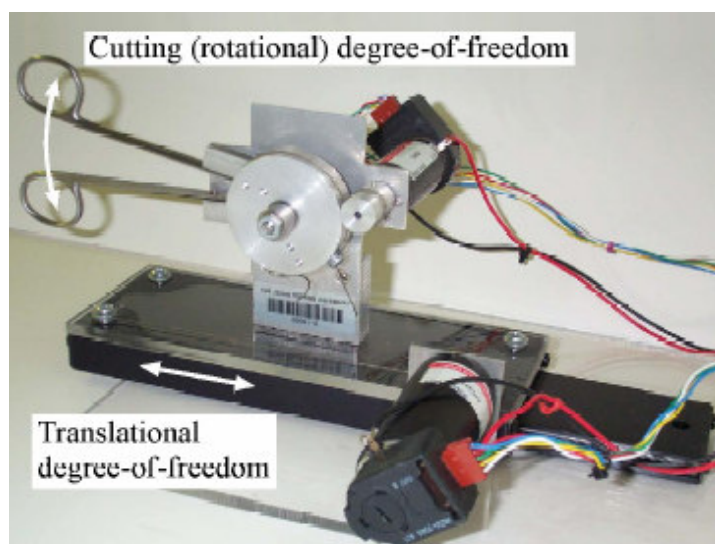


Figura 4 - Tesoura háptica com dois graus de liberdade (JAFRY *et al.*, 2003)

A Tabela 6 apresenta preços, fabricantes e tipos de equipamentos hápticos disponíveis no mercado.

Tabela 6: Preço de dispositivos hápticos (*EST - Engineering Systems Technologies, 2007 e 5DT - Fifth Dimension Technologies, 2007*)

<b>Nome do Produto</b>	<b>Fabricante</b>	<b>Preço</b>
<i>Haptic Master System</i>	<i>Moog FCS</i>	€ 42.500
<i>Delta Haptic Device 3 DOF</i>	<i>Force Dimension</i>	€ 21.144
<i>Delta Haptic Device 6 DOF</i>	<i>Force Dimension</i>	€ 38.726
<i>Omega 3 Haptic Device</i>	<i>Force Dimension</i>	€ 13.856
PHANTOM <i>Omni</i>	<i>SensAble Technologies</i>	US\$ 3,900
PHANTOM <i>Desktop</i>	<i>SensAble Technologies</i>	€ 10.000
PHANTOM <i>Premium 1.0</i>	<i>SensAble Technologies</i>	€ 16.773
PHANTOM <i>Premium 1.5</i>	<i>SensAble Technologies</i>	€ 22.727
PHANTOM <i>Premium 1.5 HF</i>	<i>SensAble Technologies</i>	€ 26.818
PHANTOM <i>Premium 3.0</i>	<i>SensAble Technologies</i>	€ 50.909
PHANTOM <i>Premium 1.5/6DOF</i>	<i>SensAble Technologies</i>	€ 45.455
PHANTOM <i>Premium 1.5/6DOF HF</i>	<i>SensAble Technologies</i>	€ 49.091
PHANTOM <i>Premium 3.0/6DOF</i>	<i>SensAble Technologies</i>	€ 66.636

Os óculos estereoscópicos, como os mostrados na Figura 5, e fabricados pela *Stereographics*, geram imagens diferentes de uma mesma cena virtual para os olhos direito e esquerdo (CELES *et al.*, 2004).



Figura 5 – Óculos estereoscópicos (*Absolute Technologies, 2007*)

Os capacetes de sistemas de RV, denominados *Head Mounted Displays* (HMDs), permitem uma imersão visual, combinando rastreadores de movimentos da cabeça e sistemas de geração de imagens. Oferecem uma visão estereoscópica (um *display* para cada olho),

incluindo, ainda, um campo de visão de 360 graus, de acordo com a orientação da cabeça, e sistemas de áudio (BERNIER *et al.*, 2004). A Figura 6 mostra um HMD (*Head Mounted Display*), construído pela *Fifth Dimension Technologies*, com resolução de 800 x 600 pixels em cada *display* (OOMS, 2004).



Figura 6 - 5DT Head Mounted Display (*Fifth Dimension Technologies*, 2007)

Algumas limitações de capacetes incluem: peso, resolução de imagens, descompasso entre o movimento visualizado pelo olho e o percebido pelo sistema vestibular, que pode causar enjoos e náuseas; limitação na geração de imagens na área periférica do olho; dificuldade de integração de sensores para rastrear posição de membros e corpo do usuário de maneira imersiva (PINHANEZ, 2004).

Existem diversos outros equipamentos que geram imagens, como VRD (*Virtual Retinal Display*), que exibem imagens diretamente na retina, óculos estereoscópicos, *displays* autoestereoscópicos, que produzem imagens estéreas por meio de monitores LCD (*Liquid Crystal Display*), e telas de plasma (BERNIER *et al.*, 2004), como o monitor mostrado na Figura 7, que apresenta imagens tridimensionais. Binóculos para geração de imagens estereoscópicas são mostrados na Figura 8.

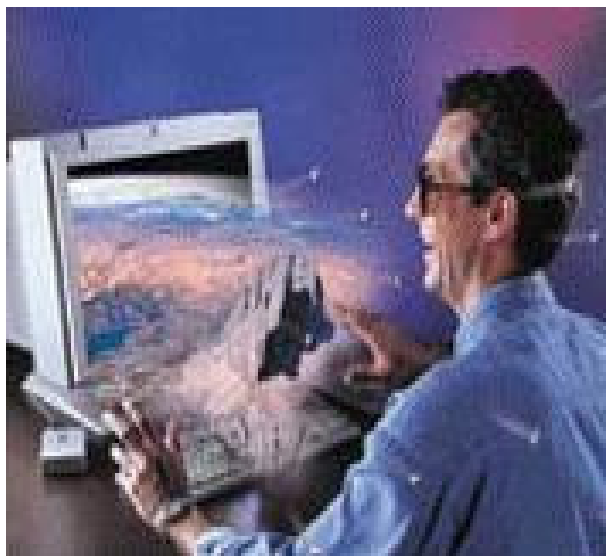


Figura 7 - Monitor ZScreen 2000i (*Fifth Dimension Technologies*, 2007)

As exibições estereoscópicas permitem uma visualização de profundidade, tornando a visão mais próxima da realidade. Neste contexto, existe o conceito de que cada um dos olhos dos seres humanos vê imagens ligeiramente diferentes, as quais são reunidas pelo cérebro, que acrescenta características de profundidade, distância, tamanho, posição (MACHADO, 2003a).

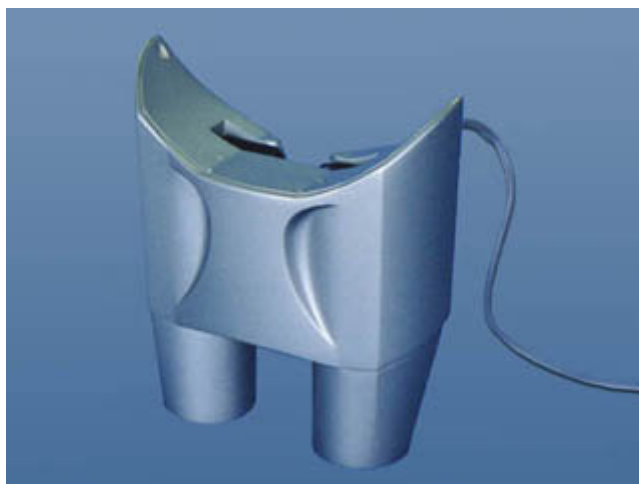


Figura 8 - 5DT Binoculars series (*Fifth Dimension Technologies*, 2007)

A Figura 9 apresenta a geração de imagens estéreas realizada por monitores LCD, fabricados pela empresa *Fifth Dimension Technologies*.



Figura 9 - Monitores estereográficos (*Fifth Dimension Technologies, 2007*)

A Tabela 7 apresenta dispositivos não convencionais disponíveis no mercado, para proporcionar visão estereoscópica, juntamente com seus preços, apresentados em moeda estrangeira, no caso, dólar americano e euro, e respectivos fabricantes.

Tabela 7: Preço de outros dispositivos não convencionais (EST - *Engineering Systems Technologies, 2007* e 5DT - *Fifth Dimension Technologies, 2007*)

<b>Nome do Produto</b>	<b>Fabricante</b>	<b>Preço</b>
5DT HMD 800-26 2D	<i>Fifth Dimension Technologies</i>	US\$ 2,995
5DT HMD 800-26 3D	<i>Fifth Dimension Technologies</i>	US\$ 3,995
5DT HMD 800-40 3D	<i>Fifth Dimension Technologies</i>	US\$ 9,995
Z800 3D Visor	<i>e-Magin</i>	€ 1.363
M920	<i>Icuiti</i>	€ 1.250
<i>i-glasses</i> PC/SVGA	<i>I-O Display System</i>	€ 817
<i>i-glasses</i> PC/SVGA PRO 3D	<i>I-O Display System</i>	€ 1.090
<i>i-glasses</i> VIDEO	<i>I-O Display System</i>	€ 899
<i>Virtual Binocular SX</i> stereo – 1280 x 1024 – 60 Hz – 40° FOV	NVIS	€ 18.091
nVisor SX - 1280 x 1024 – 60 Hz – 60° FOV	NVIS	€ 21.727
<i>ProView XL 40STm</i>	<i>Rockwell Optronics</i>	€ 50.000
<i>Arvision HMD/Goggles</i> – 800 x 600 – 40° FOV	<i>Trivisio</i>	€ 3.000

Outro dispositivo a ser citado é o *Workbench*, que consiste em uma mesa com diversos equipamentos acoplados, como óculos, dispositivo háptico, luva de dados, rastreadores de posição, projetores, câmeras, entre outros. Nesta mesa, pode-se realizar a construção de objetos no computador por meio de objetos reais (*Workbench* perceptivo) (LEIBE *et al.*, 2000), ou pode-se projetar um AV que permita ao usuário interagir com o sistema. Um *Workbench* háptico visual, como o apresentado na Figura 10, pode ser utilizado em diversas aplicações, como: simulação cirúrgica, planejamento e treinamento, prototipagem virtual e visualização científica (HANSEN *et al.*, 2003).



Figura 10 – Exemplo de um *Workbench* Háptico Visual (HANSEN *et al.*, 2003)

Em alguns sistemas, devido à exaustão causada aos usuários pelo volume de dispositivos (como determinados capacetes), sistemas de projeção foram desenvolvidos para construir AVs, como uma CAVE (*Cave Automatic Virtual Environment*), que consiste em um conjunto de telas projetadas ao redor do usuário (de 2 a 6 telas), formando um cubo, onde este pode dispor de dispositivos, tais como óculos estereoscópicos (ZIEGELER, 2002). No Laboratório de Sistemas Integráveis (LSI), da USP, há uma CAVE, que pode ser visualizada nas Figuras 11, 12 e 13, as quais apresentam a sala de projeção e a sala de controle, bem como o equipamento em funcionamento, gerando imagens nas paredes da sala.





Figura 11 - CAVE existente na USP (LSI-USP, 2007)



Figura 12 - Sala de Controle da CAVE (LSI-USP, 2007)

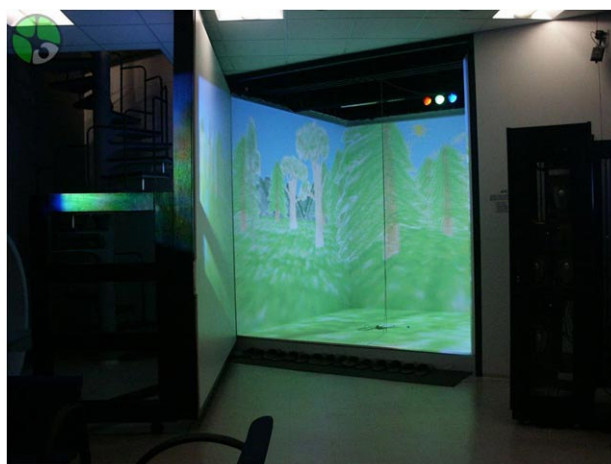


Figura 13 - Projeção nas paredes (LSI-USP, 2007)

Existem outras formas de projeção, como o Projetor *Zscreen*, fabricado pela *Fifth Dimension Technologies*, mostrado na Figura 14, que propicia a geração de imagens tridimensionais.

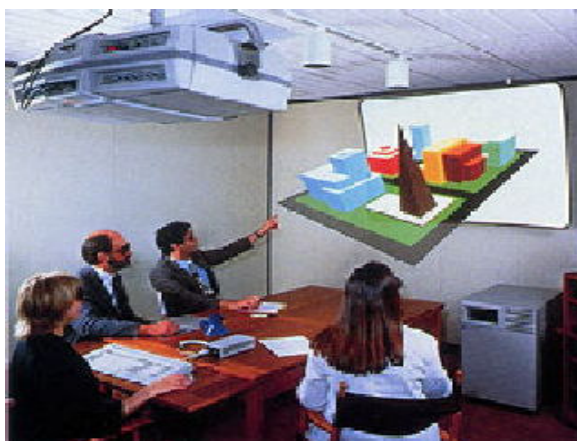


Figura 14 - Projetor Zscreen (*Fifth Dimension Technologies*, 2007)

### 1.3 Considerações Finais

A interação em AVs envolve as técnicas de interação e dispositivos. As técnicas de interação que podem ser classificadas em três grupos: navegação, seleção e manipulação, e controle do sistema que, por sua vez, possuem suas subclassificações. Os dispositivos podem ser convencionais (*mouse* comum, teclado, monitor de vídeo), ou não convencionais (luva de dados, dispositivo háptico, capacete, óculos e monitores estereoscópicos, *workbench*, *CAVE*), com suas características, limitações, custos, aplicações.

Existem também as formas de interação, que podem ser divididas em diretas e indiretas. Há uma outra classificação, que define as formas de interação como: diretas, por controles físicos e por controles virtuais.

Os AVs podem ser empregados em diversas áreas, auxiliando profissionais no desenvolvimento de suas atividades. A Medicina é uma área de interesse de RV, com diversas linhas de pesquisa, objetivando melhorar o trabalho de médicos e estudantes. A seguir são apresentadas algumas aplicações de RV na área médica, desenvolvidas em centros de pesquisa internacionais e nacionais, com destaque para a interação implementada.



## **CAPÍTULO 2 – APLICAÇÕES DE REALIDADE VIRTUAL NA MEDICINA**

Existem diversas linhas de estudo de aplicações de RV na área médica, tais como: educação médica, planejamento e simulação cirúrgica, endoscopia virtual, reabilitação e avaliação neuropsicológica (RIVA, 2003).

Na educação médica, sistemas de RV são utilizados para gerar visualizações em três dimensões e, com isso, permitir o ensino de anatomia, com a exploração de órgãos, como uma viagem pelo corpo humano. Além de uma ferramenta didática, AVs podem também ser utilizados como ferramenta de educação experimental, para testes de procedimentos médicos.

No planejamento e simulações de cirurgia, simuladores oferecem uma possibilidade do uso de computadores para treinamento de médicos, facilitando a obtenção de habilidades, uma vez que estudantes de Medicina, em geral, executam treinamentos usando objetos artificiais ou animais (RIVA, 2003). O planejamento de cirurgias consiste em identificar locais de intervenção médica, visualizar a anatomia local, verificar estruturas da anatomia e pontos sensíveis de um determinado paciente. Neste contexto, dados reais de um paciente podem ser utilizados, podendo ser obtidos por meio de tomografias computadorizadas, ultrassom, ressonância magnética, entre outros (MACHADO, 2004c). Em endoscopias virtuais, imagens tridimensionais de órgãos são geradas, permitindo que o médico visualize o procedimento e adquira conhecimento sobre este, pois tais procedimentos são denominados como invasivos, limitando a visão do cirurgião, que dispõe geralmente de uma câmera na ponta do tubo do endoscópio.

Na reabilitação e avaliação neuropsicológica, AVs permitem que um paciente controle de forma satisfatória uma situação de problema relacionada a um distúrbio, contribuindo para

sua recuperação e servindo como uma forma de médicos avaliarem suas respostas ao tratamento (RIVA, 2003).

Com a aplicação da RV na área médica, pesquisas vêm sendo realizadas no sentido de desenvolver simulações cirúrgicas, planejar operações e tratamentos radioterápicos, realizar diagnósticos mais precisos com imagens tridimensionais geradas, muitas vezes, em tempo real, e sobrepostas sobre o paciente, para facilitar o treinamento de estudantes de Medicina e substituir alguns métodos, como aqueles que consistem no sacrifício de cobaias (animais) em prol do estudo médico (COELHO e PEPPE, 2006).

## **2.1 Simulação de Coleta de Medula Óssea**

Um primeiro exemplo de aplicação de RV na Medicina neste trabalho é a extração de medula óssea. O treinamento deste procedimento é realizado geralmente em cobaias e, de acordo com Machado (2003a), apresenta desvantagens, como a ausência de visualização do que acontece dentro do corpo do animal enquanto manipula a agulha, e sacrifício de animais durante as experiências.

Diante deste cenário, pesquisadores começaram a desenvolver sistemas de RV para o treinamento desse procedimento, procurando um elevado grau de realismo nas imagens e nas sensações geradas pelo computador e dispositivos de entrada e saída, respectivamente (MACHADO, 2003a). Na Universidade de São Paulo, foi criado um sistema que simula o procedimento de extração de medula óssea, o qual consiste de um computador e um dispositivo háptico, conforme mostra a Figura 15, onde aparece um pesquisador testando o simulador (MACHADO, 2003a).



Figura 15 - Pesquisador testando simulador de extração de medula óssea (MACHADO, 2003a)

O computador gera imagens tridimensionais na tela, além de receber e enviar dados para o dispositivo háptico, que representa a agulha usada na coleta, o qual é manipulado pelo usuário.

## 2.2 Projeto VHB

Na Universidade de Ohio, Estados Unidos, pesquisadores criaram o VHB (*Virtual Haptic Back*), com o propósito de desenvolver computadores voltados a gerar simulações hápticas do corpo humano, para realizar treinamento de estudantes da área médica no que diz respeito a diagnósticos executados pela palpação de pacientes. A palpação é uma importante parte do processo de diagnóstico de muitas doenças, pois permite a detecção de tumores, hérnias e anomalias em movimentos de certos órgãos, como: coração, pulmões, intestinos e músculos. Por isso, ela é considerada uma forma efetiva, sensível e econômica para detectar diversas disfunções do corpo (BURNS *et al.*, 2004).

Entretanto, Burns *et al.* (2004) lembram que a palpação é difícil de aprender devido a três motivos:

- 1 - requer um senso de toque altamente treinado;
- 2 - estudantes, na sua maioria, praticam entre eles, os quais geralmente são pessoas jovens;
- 3 - sucessivas palpações em um sujeito podem provocar alterações no corpo, propiciando experiências diferentes para um mesmo caso.

O simulador VHB consiste de um computador responsável pela geração de imagens de alta definição de partes do corpo humano e dois dispositivos hápticos PHANTOM 3.0 (*Personal Haptic Interface Mechanism*), para que o usuário possa trabalhar com as duas mãos, conforme Figura 16, na qual um exame com um ser humano e um exame similar usando o VHB é demonstrado. Neste caso, uma imagem da coluna vertebral é utilizada, a qual foi obtida pela digitalização de um modelo da coluna vertebral com um scanner 3D (BURNS *et al.*, 2004).

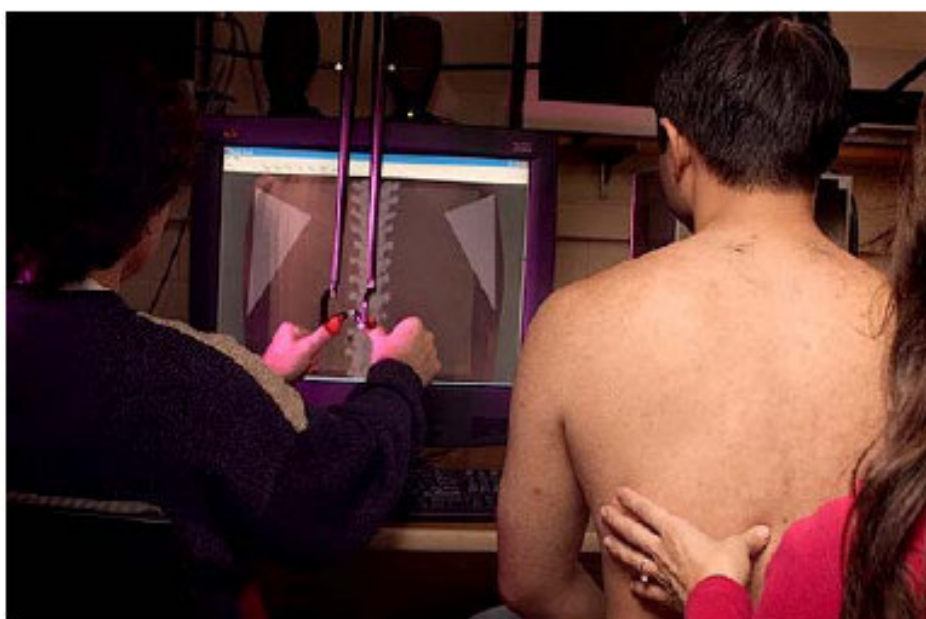


Figura 16 - VHB em operação (BURNS *et al.*, 2004)

O sistema possui também botões no topo da tela, para auxiliar o usuário e controlar a seqüência durante a interação, possibilitando a escolha da coluna vertebral, sua posição, o grau de anormalidade, bem como o nível de dificuldade, seleção e rotação de uma vértebra.

As interfaces hápticas PHANTOM 3.0, desenvolvidas pela empresa *SensAble Technologies*, usam a entrada da informação da posição definida pelo usuário para determinar quais forças retornarão ao usuário via três motores. Desta forma, o usuário move o PHANTOM com seus dedos para localizações cartesianas desejadas (x, y, z), e estas coordenadas são enviadas para o computador. Um *software* gráfico interpreta os valores e determina o vetor de força cartesiana ( $F_x$ ,  $F_y$ ,  $F_z$ ) que o usuário deverá sentir. O dispositivo gera esta força nos dedos do usuário por meio dos motores (BURNS *et al.*, 2004).

As interfaces hápticas usadas neste trabalho possuem uma resolução de 0,02 mm, um espaço de trabalho de 39x54x75 cm, e uma força aplicável de no máximo 4N. Os dispositivos somente têm a tarefa de ler as posições e aplicar forças de translação, como descrito anteriormente. Os cálculos e as imagens são gerados por um PC NT *workstation*, com processador *Pentium Xeon* de 2,8 GHZ, 1 GB de memória RAM, e uma NVIDIA Quadro4 900XGL, placa gráfica de 128 MB.

A empresa *SensAble Technologies*, fabricante dos dispositivos hápticos, oferece um kit de ferramentas de *software* denominado GHOST SDK (*General Haptics Open Software ToolKit – Software Developmente Kit*), para trabalhar com tais dispositivos, que é escrito em linguagem de programação C++ e representa o ambiente háptico como uma coleção hierárquica de objetos geométricos e efeitos espaciais. Este *kit* usa *OpenGL* e gráficos 3D, e possui algoritmo iterativo de simulação baseado em quatro funções (BURNS *et al.*, 2004):

- atualizar a posição do nó PHANTOM na cena;
- atualizar os objetos na cena;

- detectar colisões;
- enviar a força resultante para o dispositivo.

### 2.3 Simulador de Realidade Virtual para Endoscopia

Procedimentos minimamente invasivos, como por exemplo, endoscopias, são importantes em diversos tratamentos, pois possibilitam um diagnóstico mais preciso. Aparelhos usados neste tipo de procedimento são tubos flexíveis que são inseridos no sistema digestivo de pacientes, possuindo um canal óptico para transmitir imagens ao monitor do médico. Para navegação, o médico pode curvar a ponta do endoscópio em duas direções ortogonais por meio de pequenas roldanas acopladas na cabeça do aparelho, conforme Figura 17 (KORNER e MANNER, 2002).

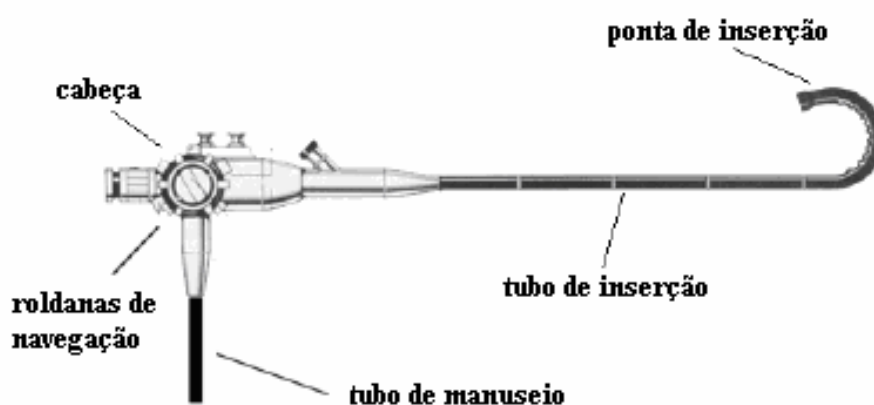


Figura 17 - Endoscópio (Adaptado de KORNER e MANNER, 2002)

Em 1999, na Alemanha, pesquisadores desenvolveram um sistema de computador para treinamento em endoscopia flexível, procurando criar um retorno de força ao usuário. Durante a sessão, o endoscópio é inserido dentro de um aparelho que representa um paciente. Então, o usuário aplica forças para mover ou rotacionar o tubo do aparelho e rotacionar as duas rodas

de navegação. Tais forças são computadas e as imagens são mostradas em um monitor, como em um exame real (KORNER e MANNER, 2002). Além disso, um endoscópio flexível tem diversos graus de liberdade, sendo que somente alguns são levados em consideração, tais como:

- translação do tubo;
- rotação do tubo;
- curvatura da ponta na direção X (que gira uma roda de navegação);
- curvatura da ponta na direção Y (que gira outra roda de navegação).

Os pesquisadores representaram forças por meio de equações calculadas pelo computador. Também construíram equipamentos com motores, correias dentadas, fios e roldanas, que imitam o funcionamento de um endoscópio. Durante a endoscopia, a visão limitada faz com que o médico precise de indícios hápticos para orientação. Quando o tubo encontra uma parede, por exemplo, a ponta do endoscópio é curvada e uma força pode ser sentida nas rodas de navegação posicionadas na cabeça do aparelho.

Para resolver este problema, a utilização de freios poderia ser necessária, com o intuito de simular a detecção de tensão no endoscópio ocasionada por um aumento da força nas rodas de navegação, que é muito comum em colonoscopias. Entretanto, somente um freio não é capaz de simular os efeitos encontrados por um médico, por isso, são necessários também *feedback* de força da ponta do tubo e *feedback* de força do próprio tubo.

Na simulação, é necessário também que o médico sinta as forças corretas na cabeça do endoscópio. Para simular a força de translação do tubo, a ponta do endoscópio foi ligada a um carro conectado a uma correia dentada, a qual é acionada por um motor, como mostra a Figura 18 (KORNER e MANNER, 2002).

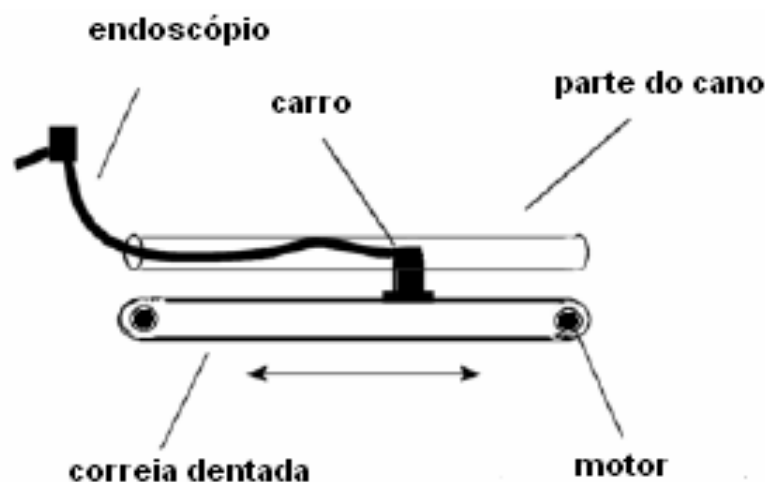


Figura 18 - Desenho de parte do simulador endoscópico (Adaptado de KORNER e MANNER, 2002)

Forças também são geradas pelos movimentos da ponta do tubo, uma vez que quando este está dentro do paciente, colisões com as paredes do cólon podem ocorrer, e o médico deve sentir tais movimentos pelas rodas do endoscópio. Para isso, foram utilizados arames dentro do tubo, entrelaçados ao redor de parafusos acionados por dois motores e ligados a parafusos na outra extremidade do tubo para acionar as roldanas de navegação, presentes na cabeça de um endoscópio real (KORNER e MANNER, 2002).

## 2.4 Simulador para Treinamento de Acupuntura Chinesa

Outro exemplo da utilização de dispositivo háptico em sistemas para treinamento médico, é um simulador de manipulação de agulha, que tem como propósito auxiliar na aprendizagem e treinamento de acupuntura chinesa. Esta técnica é empregada no tratamento e prevenção de diversas doenças, na qual por meio da inserção e extração de agulhas é feita a estimulação de determinados pontos do corpo humano, controlando a frequência e a profundidade.



Por isso, a precisão e a correta manipulação das agulhas são determinantes para um tratamento satisfatório, exigindo habilidade e experiência. Os estudantes de acupuntura geralmente treinam em pacientes reais ou manequins artificiais, sendo que nos primeiros, o procedimento pode causar dores se mal executado, e os segundos não oferecem uma experiência real. Um sistema com um modelo humano virtual e um dispositivo que propicie um retorno de força para a tarefa de inserção, pode oferecer informações úteis aos estudantes, com um certo grau de realismo e sem prejudicar a saúde de pacientes (CHUI *et al.*, 2004).

O simulador apresentado por Chui *et al.* (2004) oferece ao usuário um menu principal e, dentre as funções mais importantes, citam-se o uso da agulha e o ajuste de modelo. Selecionando a primeira função, as sessões de treinamento são iniciadas com a escolha aleatória de pontos do corpo humano, os quais são apresentados por meio de um monitor ao estudante, juntamente com um objeto virtual representando uma agulha. O usuário deve inserir a agulha nesses pontos, manuseando um equipamento háptico de seis graus de liberdade, como mostra a Figura 19. O estudante pode alterar a posição e orientação do corpo virtual, solicitar informações sobre os pontos e sobre seu desempenho no treinamento ao final de uma sessão, além de manter a agulha virtual em uma determinada posição, quando esta está dentro do corpo humano virtual (CHUI *et al.*, 2004).

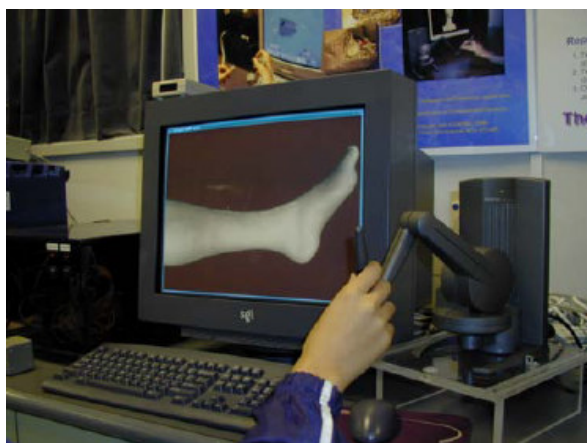


Figura 19 - Simulador de Acupuntura Chinesa em execução (CHUI *et al.*, 2004)

A segunda opção permite o ajuste de parâmetros de controle do modelo háptico, divididos em onze etapas, os quais podem ser definidos de acordo com a experiência dos participantes da simulação. Os valores dos parâmetros podem ser armazenados pelo sistema e utilizados em outras sessões.

Para a construção do sistema, optou-se pela separação do processamento de imagens e do processamento háptico, com o objetivo de atingir uma eficiência elevada. Desta forma, o simulador de acupuntura chinesa é constituído de duas máquinas para processamento, sendo uma *SGI Octane2 workstation*, responsável pela renderização das cenas durante o treinamento, denominada de servidor visual, e uma *Windows NT workstation*, conectada à primeira por meio de uma rede e responsável pelo recebimento e envio de informações ao dispositivo *PHANTOM Desktop*, denominada de servidor háptico.

O *overhead*<sup>4</sup> da rede foi considerado durante os testes. Entretanto, os autores afirmaram que a eficiência global foi satisfatória, alcançando taxas de atualização de imagens em frequências acima de 30 Hz, e taxas de atualização de retorno háptico em torno de uma frequência de 1000 Hz. A renderização visual foi programada em linguagem C++ com suporte das bibliotecas *Visualization Toolkit* e *GIMP Toolkit*. A renderização háptica foi também programada em linguagem C++, com auxílio da biblioteca *GHOST*. Antes da construção de uma cena, o servidor visual envia um sinal ao servidor háptico, solicitando a posição e orientação do dispositivo para definir a agulha virtual na tela que, por sua vez, envia a resposta somente depois da solicitação, já que trabalha em uma frequência maior, o que poderia causar problemas de sincronização. Além dessas mensagens trocadas entre os servidores, o servidor visual envia informações quando o teclado ou o *mouse* são acionados, indicando mudanças no modo de interação (CHUI *et al.*, 2004).

---

<sup>4</sup> Excesso na utilização de um recurso computacional.

Na construção do modelo háptico, alguns fatores foram levados em consideração, como: realismo, procurando simular o comportamento de força na manipulação de agulhas durante uma sessão de acupuntura; computação, buscando uma eficiência computacional que atinja uma taxa de atualização aceitável; definição de equações para cálculo de força e limitações, envolvendo o equipamento e suas limitações com relação ao máximo de força e tenacidade.

Nesse contexto, as categorias de tecido do corpo humano foram estudadas e modeladas, como pele, tecido adiposo, músculo e osso, definindo equações que simulassem o comportamento da agulha quando em contato com esses tecidos. Para isso, foram consideradas características, como viscosidade, elasticidade, deformação, fricção entre agulha e tecido, e tenacidade do tecido. Além disso, foi necessário calcular a composição de forças conforme a agulha atravessa as camadas de tecido.

## **2.5 Sistema Interativo de Treinamento em Exame Ginecológico**

Baseando-se nas vantagens oferecidas pelos sistemas de RV voltados à área médica, pesquisadores construíram um simulador com o objetivo de disponibilizar uma ferramenta para a identificação de doenças relacionadas ao colo do útero, denominada SITEG (Sistema Interativo de Treinamento em Exame Ginecológico), que oferece diferentes patologias e estágios, disponibilizando informações sobre as patologias, avaliando o desempenho do usuário e permitindo a realização das duas etapas do exame. O simulador oferece casos de colo normal, Herpes e HPV (*Human Papiloma Virus*), com interação em tempo real, geração de imagens estereoscópicas e dispositivo háptico para sensação de toque (MACHADO *et al.*, 2004e).

O exame do colo do útero é realizado em duas etapas: uma visual, que consiste na observação de características da vagina e do colo uterino por meio de um espéculo<sup>5</sup>, e uma tátil, que consiste no toque do canal vaginal e do colo uterino, para avaliar as características do tecido.

O desenvolvimento do sistema se baseou em quatro abordagens: construção dos modelos, geração e controle de visualização da cena, geração e controle de sensações táteis, e sistema de avaliação, que foram categorizadas em classes (MACHADO *et al.*, 2004e).

Para a execução foram criados dois módulos:

1 - Estudo, com a escolha de um caso ou patologia por parte do usuário, para apresentação de informações visuais e táteis relacionadas a escolha feita; e

2 - Diagnóstico, que permite o treinamento da capacidade do usuário de diagnóstico, sendo a escolha realizada pelo SITEG de forma aleatória. O sistema apresenta ainda submódulos: de avaliação, conectado ao módulo de diagnóstico, que compara informações do treinamento do usuário com informações de um diagnóstico correto; visual, que mostra um modelo tridimensional de um espéculo; toque, que apresenta um modelo da genitália externa e do dedo do participante, sendo este último acionado pelo equipamento háptico. Tanto para os módulos quanto os submódulos, quatro opções de visualização podem ser escolhidas, sendo uma monoscópica e três estereoscópicas, permitindo a visualização de imagens em três dimensões.

No que diz respeito à plataforma de execução do sistema, com o intuito de reduzir custos, optou-se pela plataforma *Linux*. A interação entre usuário e sistema dá-se por meio de um *mouse* convencional, de um teclado, que ativam e desativam opções de um menu, e um equipamento háptico. Este dispositivo no submódulo visual é utilizado para posicionar a fonte

---

<sup>5</sup> Instrumento em forma de tubo que permite a observação de determinadas cavidades do corpo.

de luz e iluminar o canal vaginal, e no submódulo tátil oferece um retorno tátil e de força, tornando a interação intuitiva e próxima da realidade. A linguagem de programação escolhida foi C++, em conjunto com a biblioteca *OpenHaptics*, que disponibiliza funções e comandos para controle do dispositivo háptico, além de serem compatíveis com o *OpenGL*.

Para calibração do sistema, foram usadas informações fornecidas por um médico experiente em diagnósticos desta natureza, e foram traduzidas numericamente para processamento. Além disso, um médico também testou o sistema, definindo propriedades de elasticidade e textura, que foram adicionadas aos modelos (MACHADO *et al.*, 2004e).

## 2.6 Considerações Finais

Os AVs podem ser projetados para auxiliar profissionais de diversas áreas da saúde, como médicos e estudantes de Medicina, permitindo a realização de treinamentos com diferentes níveis de dificuldade e situações, minimizando a necessidade de testes em pacientes reais, cadáveres e animais. Podem, ainda, evitar constrangimento dos pacientes, experiências mal sucedidas e minimizar o receio de cometer erros por parte dos estudantes e médicos.

Neste contexto, a escolha de dispositivos é muito importante, uma vez que envolve diversas questões, como disponibilidade financeira e grau de realismo proporcionado. Dispositivos convencionais como teclado e *mouse* são mais baratos que os não convencionais como luva de dados e equipamento háptico, entretanto, estes últimos oferecem um grau de realismo mais elevado em uma simulação, pois assemelham-se a objetos do mundo real.

Além do citado, outra questão importante no desenvolvimento de aplicações dessa natureza, é a avaliação de tais aplicações, o que permite a definição de parâmetros para a construção ou aperfeiçoamento das mesmas, tornando-as mais próximas da realidade, e

conseqüentemente, mais confiáveis no auxílio dos profissionais da área médica, visto que a interação deve propiciar experiências e conhecimento satisfatórios.

A avaliação de sistemas de RV no contexto considerado neste trabalho é o assunto abordado no próximo capítulo.

## CAPÍTULO 3 - AVALIAÇÃO DE INTERAÇÃO EM AVs

Para o projeto e desenvolvimento de interfaces que promovam a interação entre homem e máquina, devem ser levados em consideração tanto aspectos computacionais, os quais envolvem dispositivos de comunicação e arquitetura de *software*, quanto aspectos relacionados à natureza humana, envolvendo questões físicas, fisiológicas, psicológicas, entre outras, procurando facilitar o uso de sistemas, além de elevar o nível de aprendizado e a motivação (BAECKER *et al.*, 2004).

Os AVs podem possuir interfaces avançadas, tornando o projeto destas uma tarefa complexa. Entretanto, abrem um horizonte de novas possibilidades, uma vez que tais ambientes oferecem interfaces elaboradas com gráficos tridimensionais e técnicas de interação em três dimensões (KAUR *et al.*, 1998).

Em sistemas de RV, para manter a imersão e um grau adequado de realismo, é necessário, primeiro, que o computador processe ou renderize uma determinada quantidade de quadros a ser exibida por segundo, e segundo, que as imagens sejam de alta qualidade (HADWIGER, 2007). Um sistema de RV envolve uma determinada quantidade de polígonos para compor uma cena, resolução de texturas, número de usuários, tempo de renderização, bem como características da tarefa a ser executada e número de objetos no AV (BOWMAN *et al.*, 1999). Entretanto, além dos aspectos computacionais envolvidos na geração de um AV, como dispositivos empregados, máquinas para processamento, redes de computadores e sistemas distribuídos, uma avaliação também deve conter informações sobre os participantes da interação, ou seja, o usuário.

Os desenvolvedores de AVs, em geral, se preocupam com a implementação de dispositivos e a criação de técnicas de interação, mas muitas vezes deixam de lado a questão dos usuários, os participantes da interação no ambiente sintético (GABBARD e HIX, 2002). Por isso, estudos vêm sendo realizados com o intuito de identificar os problemas de usabilidade de um projeto de interface ou de uma interface implementada, causando a necessidade de criação de métodos e parâmetros para realizar avaliações.

A usabilidade consiste na facilidade de uso e utilidade de um sistema, incluindo satisfação do usuário, capacidade de aprendizado, velocidade e acurácia na execução de uma determinada tarefa (BOWMAN *et al.*, 2001a). A Engenharia de Usabilidade procura analisar o usuário e as tarefas, elaborar projetos de interação, construir protótipos e métodos de avaliação.

Para realizar a avaliação de sistemas, deve-se criar uma lista de fatores, que são características mensuráveis de interação em um AV. Um exemplo, citado por BOWMAN *et al.* (2001a), é o conjunto de fatores de qualidade desenvolvido em Atlanta, Estados Unidos, para uma técnica de *travel* imersiva, que inclui:

- **velocidade:** define um ritmo ideal de navegação;
- **acurácia:** consiste na aproximação do objetivo que se deseja atingir;
- **consciência espacial:** conhecimento do usuário com relação a sua posição e orientação no AV;
- **facilidade de aprendizado:** trata da habilidade de um novo usuário em utilizar o sistema;
- **facilidade de uso:** compreende as dificuldades de execução da técnica sob o enfoque do usuário;



- **ganho de informação:** engloba a habilidade do usuário na obtenção de informações do ambiente durante a interação;

- **presença:** compreende o senso de imersão;

- **conforto do usuário:** expressado pelo próprio usuário, no que se refere aos sentimentos de vertigem, náuseas, stress, entre outros, durante a navegação (BOWMAN *et al.*, 2006).

Um método ou uma técnica de avaliação desejável deve identificar problemas de usabilidade em um sistema, ser de fácil utilização, apresentar resultados precisos e confiáveis e não necessitar de avaliadores experientes para execução (BARCIA *et al.*, 2007).

### 3.1 Fatores Humanos de Interação

Como mencionado, fatores humanos são significativos para avaliação de interação entre homem e computador, envolvendo capacidades e limitações do usuário, e podendo ser divididos em três áreas: eficiência do desempenho humano, questões de segurança e saúde, implicações sociais. Dentro deste contexto, deve ser realizado o entendimento de algumas questões, como: quais características do usuário podem afetar o desempenho do AV, possíveis problemas de saúde no usuário devido à exposição ao AV, limitações do AV percebidas pelo usuário, tarefas realizadas de forma satisfatória e nível de entendimento da tarefa e do uso do sistema por parte do usuário (KENNEDY *et al.*, 1998).

Nesta área da eficiência do desempenho do participante, o objetivo é definir princípios que maximizem o desempenho do usuário na execução de uma determinada tarefa, englobando complexidade de navegação e grau de presença, bem como características da tarefa e do usuário, interação de múltiplos modos, formas de visualização, audição e retorno

háptico, além de limitações sensoriais e motoras dos indivíduos. Na questão de saúde e segurança, são incluídos efeitos diretos, os quais são classificados em microscópicos (olhos afetados por campo eletromagnético durante exposição prolongada a equipamento de exibição, por exemplo) e macroscópicos (ocorrência de traumas, por exemplo); e efeitos indiretos, que consistem em distúrbios psicológicos. O impacto social causado pela tecnologia trata de problemas relacionados ao comportamento na vida real influenciado por interações em AVs, principalmente naqueles que apresentam cenário de violência e destruição (KENNEDY *et al.*, 1998).

Os dispositivos tornam a interação mais intuitiva e eficiente (GRANTZ e ROESSLER, 1998). Por isso, um estudo para avaliar um simulador de treinamento médico com equipamento háptico acoplado é apresentado na seção seguinte, com a coleta de informações sobre retorno háptico, tempo de processamento e percepção do usuário (relação entre os retornos háptico e visual), uma vez que determinadas interações envolvem visualização e retorno háptico, exigindo processamento para geração de imagens, além de cálculo de força e envio de resposta ao dispositivo.

### **3.2 Avaliação de Dispositivos Hápticos em Simulações Cirúrgicas**

Como já mencionado, estudantes de Medicina precisam de experiência para realizar procedimentos médicos. Geralmente o treinamento é realizado em animais, cadáveres e pacientes reais. No entanto, animais não possuem a mesma anatomia que humanos e cadáveres não possuem uma correta resposta fisiológica. Em pacientes reais há um risco do estudante não ter a competência necessária para executar o procedimento, podendo gerar insegurança e riscos ao paciente.

Um AV pode constituir uma solução adequada, e um *feedback* háptico é indispensável para um treinamento altamente realístico em simuladores cirúrgicos. De acordo com BATTEAU *et al.* (2004), pesquisadores estão estudando a percepção háptica desses simuladores, na qual um dos problemas encontrados é a latência, o atraso entre a ação do usuário e a resposta do sistema, uma vez que isto pode prejudicar o treinamento, já que é necessária uma determinada precisão.

Na pesquisa realizada na Universidade de *Utrecht*, na Holanda, dois experimentos foram realizados para a percepção da latência e o retorno háptico humano, o que pode ser considerada como uma tarefa complexa, pois a habilidade humana para sentir a latência háptica é pobre (BATTEAU *et al.*, 2004).

Para os experimentos, foi utilizado um dispositivo háptico *CathSim AccuTouch*, conectado a um PC *workstation*. O dispositivo, conforme mostrado na Figura 20, consiste de uma haste representando uma agulha, acoplada a um freio magnético, sendo que a haste ou agulha tem três graus de liberdade de movimento, que podem ser detectados e processados pelo PC (BATTEAU *et al.*, 2004).

O freio magnético, mostrado também na Figura 20, oferece um grau de liberdade com ou sem um *feedback* háptico de força e reflexão, à medida que a agulha é inserida ou extraída, sendo possível a aplicação de aproximadamente 4N de força de resistência.



Figura 20 - Dispositivo Háptico *CathSim* (BATTEAU *et al.*, 2004)

Um modelo semelhante ao *CathSim AccuTouch* é mostrado na Figura 21a, na qual pode-se observar a agulha a ser manipulada e o freio magnético. A Figura 21b apresenta o uso deste modelo (IMMERSION, 2008).



Figura 21a – Modelo de *CathSim* (IMMERSION, 2008)

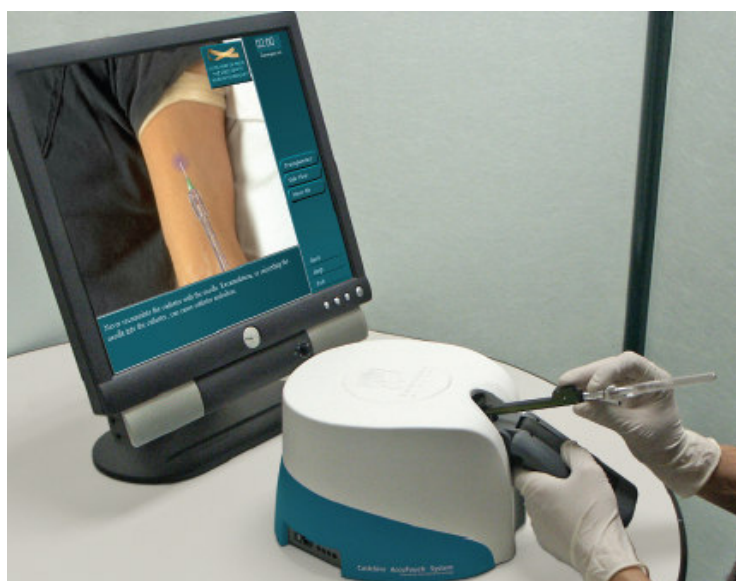


Figura 21b – Treinamento com o modelo (IMMERSION, 2008)

No trabalho desenvolvido por BATTEAU *et al.* (2004), na tela do computador, o usuário visualiza uma agulha que é inserida na pele, causando uma deformação. Esta deformação é calculada e mostrada na imagem, conforme apresentado na Figura 22. Durante a inserção da agulha é percebida uma resistência, transmitida pelo aparelho para a mão do usuário. Quando a pele é perfurada, o dispositivo simula um estouro. Para calcular os efeitos da inserção e extração da agulha, utilizou-se a equação 1.

$$FG = \begin{cases} FG_p \left( \frac{x}{x_p} \right)^2, & \text{se } 0 < x < x_p \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

onde  $FG \in [0, 1]$  é o ganho de força, que é a fração máxima que o dispositivo é capaz de alcançar, próximo de 4N;  $x$  é a distância entre a ponta da agulha e o ponto de inserção na pele, e  $x_p$  é o ponto de punção. Durante a extração,  $FG$  será zero até a agulha ser retirada da pele.



Figura 22 - Deformação virtual da pele durante procedimento (BATTEAU *et al.*, 2004)

Estudantes de Medicina, médicos e enfermeiros testaram o sistema conforme descrito a seguir. No primeiro experimento o objetivo era determinar a consistência do retorno de uma percepção háptica. Resultados desses testes podem oferecer uma percepção dentro de uma precisão necessária para *feedback* háptico em simuladores médicos. Os testes contaram com vinte e sete voluntários com experiências diferentes. Como o foco era a consistência, não foi determinada qual simulação cada um deveria realizar, sendo permitida a execução de qualquer procedimento desejado pelo usuário. Os voluntários foram instruídos a ajustar a função matemática (equação 1), até que a resposta háptica fosse consistente. Depois, o experimento com o valor ajustado foi repetido dez vezes (BATTEAU *et al.*, 2004).

No segundo experimento, realizado com vinte e oito voluntários, a intenção foi investigar a habilidade para detectar a latência entre o *feedback* visual e *feedback* háptico por meio de uma variável. Cada voluntário começou com uma latência na faixa de 120-150 ms, que foi gradualmente decrementada em 15 ms até que o voluntário dissesse que a latência poderia não ser percebida. Este valor foi anotado e o teste reiniciado com uma latência menor, em uma faixa de 0-30 ms. Então, o valor foi gradualmente incrementado até que o voluntário dissesse que ele estava percebendo a latência. O segundo valor também foi anotado e o experimento repetiu-se por três vezes (BATTEAU *et al.*, 2004).

Para cada voluntário foram calculados a média e desvio-padrão da função empregada, sendo que a primeira indicou uma seleção consistente de resistência da agulha com o procedimento simulado, e o segundo indicou uma consistência na resposta do usuário. O primeiro é dependente da suposição do procedimento de inserção, e o segundo depende da resposta do usuário e a habilidade para detectar variações no *feedback* háptico. A métrica escolhida para analisar os dados foi a Lei de Weber (equação 2), com o objetivo de normalizar a consistência dos voluntários.

$$W = \frac{\text{desvio-padrão}}{\text{média}} \quad (2)$$

Sendo assim, um valor pequeno de  $W$  indica que o voluntário é capaz de reproduzir o mesmo nível de *feedback* háptico, e um valor alto de  $W$  indica uma consistência pobre. Quatorze por cento dos 27 voluntários apresentaram valores de  $W$  mais altos do que o restante do grupo, significando que a habilidade não depende da experiência, já que os níveis de experiência eram diversificados (BATTEAU *et al.*, 2004).

Para cada voluntário foi calculada também uma média de latência com base nos valores de latência mais altos e mais baixos. Para todos os participantes, a média e o desvio-padrão foram em torno de 98 ms e 19 ms, respectivamente. De acordo com os pesquisadores,

uma distribuição acumulativa da latência mostrou que 99% são incapazes de detectar uma latência de 54 ms entre os *feedback* visual e háptico.

Os autores afirmaram, ainda, que os experimentos demonstraram que a consistência de respostas hápticas pode variar amplamente de pessoa para pessoa. A métrica W oferece uma comparação entre os participantes dos testes, e como 99% das pessoas não percebem uma latência de 54 ms, este tempo pode ser utilizado para processamento e refinamento do *feedback* háptico.

### **3.3 Avaliação de Técnicas de Seleção e Manipulação em Mundo Virtuais**

Como já descrito, as operações de navegação, seleção e manipulação de objetos no AV são baseadas em técnicas de interação, as quais estão ligadas com os chamados dispositivos de entrada e saída. Nesta seção será apresentado um trabalho realizado na Universidade do Rio Grande do Sul, que teve como objetivo avaliar técnicas de seleção e manipulação em um AV imersivo, que representa um jogo de xadrez, cujas técnicas avaliadas foram: mão virtual, associada a um *mouse* convencional e a uma luva de dados, e *ray-casting*, associada a luva de dados (FREITAS *et al.*, 2002).

A aplicação consiste de um jogo de xadrez, sendo o AV constituído de um tabuleiro com 64 casas e 32 peças. O usuário pode visualizar o tabuleiro de duas formas: uma vista superior, simulando uma interface bidimensional, e uma com inclinação de 60 graus, buscando uma proximidade com a realidade. Ao usuário não é permitido movimentar peças para fora do tabuleiro ou selecionar mais de uma peça simultaneamente. É permitido movimentar uma peça por vez, e para uma nova seleção é preciso que todas as peças estejam no tabuleiro. Na movimentação de uma peça, esta é suspensa sobre o tabuleiro, e a troca de posição pode então ser realizada.

FREITAS *et al.* (2002) citam que as técnicas de interação podem ser classificadas de diversas formas, inclusive como metáforas de interação. Estas por sua vez podem ser categorizadas em egocêntricas, na qual o usuário é tido como parte do mundo virtual e a proporção entre objetos e usuário deve ser levada em consideração; e exocêntricas, nas quais as proporções entre objetos e usuário não são consideradas (POUPYREV, 1998 **apud** FREITAS *et al.*, 2002).

No trabalho citado foi utilizada a metáfora egocêntrica, visto que a exocêntrica seria mais adequada em ambientes onde os objetos se encontram dispersos e, em um jogo de xadrez, as peças estão ao alcance da mão do jogador. Na técnica da mão virtual, a mão do usuário é representada no AV, e ele altera a posição e a orientação por meio de um dispositivo de entrada, como uma luva de dados. Para a seleção, o participante deve posicionar a representação da mão dentro do objeto desejado e informar a sua intenção, utilizando um botão. Na técnica *ray-casting*, um raio infinito e semi-transparente parte da representação da mão virtual, sendo sua direção determinada pela orientação desta. Para a seleção, o usuário aponta para o objeto e novamente informa o seu desejo.

Além disso, durante os testes, situações foram consideradas com relação a ausência e presença de *feedback* visual, sendo que neste último caso a peça selecionada se torna vermelha e é suspensa no tabuleiro, indicando a seleção. Outra consideração foi a respeito da visualização, que pode ser realizada em 0 grau ou em uma inclinação de 60 graus, permitindo nesta última uma noção melhor de distância entre as peças e o usuário, entretanto, necessitando de visão esteresocópica, o que aumenta o grau de imersão.

Para o estudo realizado foram empregados quatro dispositivos: *mouse*, luva de dados, rastreadores de movimento e óculos de cristal líquido. O *mouse* convencional foi usado para guiar a mão virtual nas tarefas de seleção e manipulação (FREITAS *et al.*, 2002).



A luva adotada foi uma 5DT *Data Glove 5*, que capta diversos gestos do usuário, uma vez que possui um sensor de rotação e inclinação, bem como um sensor de flexão (fibra óptica) para cada dedo, sendo a resolução da flexão de cada dedo de 8 bits, o que possibilita 256 posições distintas, e a resolução de rotação e inclinação de -60 a 60 graus. Foi utilizada na implementação de interação com as técnicas da mão virtual e *ray-casting*. Neste contexto, a seleção é realizada com um gesto de fechar da mão do usuário, o movimento com a mão fechada significa mover a peça, e o gesto de abrir da mão indica o ato de soltar a peça em uma determinada posição do tabuleiro.

Outro equipamento utilizado foi um rastreador de movimento, do tipo *Flocks of Birds*, comercializado pela empresa *Ascension Technologies*, com capacidade para medir um campo magnético emitido por uma fonte, gerando posição e orientação no espaço tridimensional, pois a luva não era capaz de captar movimentos de translação. Tal equipamento é constituído de fonte geradora e um sensor que é colocado na mão do usuário, sobre a luva. Para visualização, foram escolhidos óculos de cristal líquido denominados *H3D Glasses*, que geram imagens para os olhos direito e esquerdo a uma taxa de 30 imagens por segundo.

A metodologia para avaliação de seleção e manipulação adotada baseou-se em cinco hipóteses, compostas por variáveis independentes que seriam alteradas para testar diferentes hipóteses, e variáveis dependentes, que mediriam o desempenho dos usuários. As hipóteses estão ligadas a determinadas preferências dos usuários, sendo (FREITAS *et al.*, 2002):

1 – O usuário prefere o tabuleiro a 0 grau e terá problemas com o tabuleiro inclinado, pois ele está mais familiarizado com interfaces bidimensionais;

2 – O *feedback* visual influencia no desempenho dos usuários na interação, uma vez que existem diversos objetos no ambiente, este tipo de resposta pode ser útil na identificação do objeto a ser selecionado e manipulado;

3 – A técnica da mão virtual será a preferida dos usuários quando o dispositivo de entrada for a luva de dados, já que a tarefa realizada no AV é semelhante à tarefa realizada no mundo real, e a técnica de *ray-casting* apresenta baixo desempenho para distanciar e aproximar objetos (BOWMAN, 1997; POUPYREV, 1998 **apud** FREITAS *et al.*, 2002);

4 – O dispositivo preferencial será o *mouse*, pois os usuários estão habituados com este equipamento;

5 – O desempenho do usuário será mais elevado quando o tabuleiro estiver em 0 grau e a técnica a ser utilizada for a *ray-casting*, uma vez que esta é considerada uma técnica bidimensional e o tabuleiro apresentará uma interface bidimensional.

As variáveis independentes consistem nas características do sistema que podem ser manipuladas para produção de diferentes condições de comparação. Neste estudo foram definidas as seguintes variáveis independentes (FREITAS *et al.*, 2002):

- **inclinação do tabuleiro (0 e 60 graus):** permitindo testar a hipótese 1 e auxiliando no teste da hipótese 5;

- **feedback visual, para testar a hipótese 2:** tendo como *feedback* a cor da peça selecionada;

- **dispositivo utilizado:** envolvendo a utilização de luva de dados e *mouse* e permitindo o teste das hipóteses 3, 4 e 5. Quando o dispositivo for o *mouse*, o cursor torna-se um objeto que representa uma mão, relacionando-se diretamente com a hipótese 4. Quando for a luva de dados, o tipo de cursor depende da técnica de interação (mão virtual e *ray-casting*);

- **técnica de interação com a luva:** relacionada com a hipótese 3, que envolve a mão virtual, e a hipótese 5, que combina a orientação do tabuleiro e a técnica de *ray-casting*.

As variáveis dependentes por sua vez, são medidas usadas para indicar o desempenho ou a aceitação da técnica por parte dos usuários, e podem ser objetivas, como o tempo para selecionar e manipular um objeto, ou subjetivas, as quais são coletadas por meio de questionários. Neste estudo, foram adotadas as seguintes variáveis (FREITAS *et al.*, 2002):

- **eficiência da técnica de interação:** definindo o tempo de realização de uma determinada tarefa;
- **facilidade de uso da técnica de interação:** relacionando-se com a simplicidade de uso de cada técnica;
- **conforto/desconforto do dispositivo utilizado:** medindo o conforto em relação a cada dispositivo, levando em consideração o peso do braço, da mão e do sentimento de náuseas durante a interação;
- **apreciação pessoal do usuário:** levando em conta a preferência do usuário por um determinado dispositivo.

O experimento consistiu na execução de movimentos determinados com as peças do jogo de xadrez, com contabilização do tempo de execução dos movimentos e da quantidade de erros até a finalização de cada tarefa, além da aplicação de questionários.

Para realizar as tarefas foram selecionadas 29 pessoas, sendo que 27 não possuíam familiaridade com os equipamentos não convencionais. As tarefas foram definidas da seguinte forma: peças deveriam ser movimentadas no eixo x e no eixo z, realizando a seleção e o deslocamento para uma determinada posição; cada pessoa realizaria 24 tarefas, sendo 12 para movimentação de peças no eixo x e 12 para movimentação no eixo y; 8 tarefas seriam executadas com o *mouse* e 16 com a luva de dados, com este último dispositivo seriam implementadas as duas técnicas descritas anteriormente; em 12 tarefas as peças sofreriam alterações em sua coloração (*feedback* visual), e em outras 12 não haveria nenhuma

modificação; em 12 tarefas o tabuleiro estaria inclinado, e nas outras 12 não haveria inclinação. Cada usuário teria de 5 a 10 minutos para realizar a tarefa, recebendo mensagens de instruções na tela do computador sobre a peça a ser manipulada, o dispositivo a ser utilizado e o movimento a ser realizado.

Um arquivo no sistema gravava os tempos de seleção e manipulação, contendo nome do usuário, número de cliques no vazio quando o *mouse* era utilizado, peças selecionadas de forma errada e quantidade de manipulações erradas quando as peças eram colocadas na posição errada (FREITAS *et al.*, 2002).

Baseando-se nos tempos registrados, nas informações coletadas com o questionário, e analisando as diferenças entre os tempos médios, FREITAS *et al.* (2002) concluiu-se de acordo com as hipóteses levantadas anteriormente o seguinte: para a hipótese número um, sobre a preferência pelo tabuleiro sem inclinação, notou-se que, mesmo a maior parte dos usuários ter declarado preferência pelo tabuleiro inclinado em 60 graus, não houve diferença significativa entre os tempos médios quando a seleção ocorreu no eixo x, entretanto, quando a seleção ocorreu no eixo z, os tempos foram menores com o tabuleiro a 0 grau. No caso dos tempos de manipulação, o resultado foi o contrário, com os usuários tendo um melhor desempenho quando o tabuleiro estava inclinado em 60 graus para movimentos no eixo x, e não havendo diferença significativa na manipulação realizada no eixo z.

No que diz respeito à segunda hipótese, mesmo as pessoas declarando que o *feedback* visual ajudou na realização das tarefas, os tempos médios para ambas as tarefas (seleção e manipulação), com e sem *feedback* não demonstraram diferenças significativas. Sobre a terceira hipótese, com a mão virtual sendo considerada a técnica desejável quando da utilização da luva, houve a confirmação de que esta técnica apresenta um desempenho melhor nas tarefas de seleção e manipulação no eixo z. No entanto, no eixo x o desempenho foi

semelhante para ambas as técnicas. Em se tratando da quarta hipótese, adotando o *mouse* como dispositivo preferido, houve a confirmação da hipótese na avaliação subjetiva e nos tempos médios de seleção e manipulação, com tempos significativamente menores do que aqueles conseguidos com a luva de dados.

A última hipótese foi confirmada (usuário com melhor desempenho quando utiliza a técnica de *ray-casting* e o tabuleiro está a 0 grau), com os tempos médios para as tarefas de seleção significativamente menores do que aqueles realizados com o tabuleiro a 60 graus. Entretanto, com relação aos tempos de manipulação houve diferenças significativas somente na movimentação no eixo x, quando o tabuleiro estava a 0 grau.

### **3.4 Uma Avaliação Sobre o Uso de Estímulos Táteis em um AV**

Um trabalho realizado na Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), procurou avaliar a utilização de um dispositivo que gerava estímulo tátil em um AV. Os autores tinham o objetivo de construir equipamentos com custo acessível baseados em micro-motores vibratórios, encontrados em aparelhos celulares.

Foi realizada uma avaliação de usabilidade em um AV imersivo, com aplicação de testes em cada etapa do processo interativo e a utilização de um jogo de quebra-cabeça virtual, para coleta de dados necessários na análise (KOPPER *et al.*, 2006).

O sistema de interação foi constituído por diversas tarefas, agrupadas de acordo com características semelhantes, definindo cada etapa do processo interativo. Os autores definem que o processo de manipulação de objetos virtuais, por exemplo, pode ser dividido em quatro etapas (BOWMAN, 1999 **apud** KOPPER *et al.*, 2006):

- **seleção:** define o que ocorre quando o usuário está tentando localizar um objeto para manipular;

- **anexação:** consiste no que ocorre no momento que o usuário confirma o objeto a ser selecionado e anexa este ao seu apontador;

- **posicionamento:** trata das ações que devem ocorrer quando o objeto está sendo movimentado pelo usuário;

- **liberação:** especifica o que ocorre no momento que o objeto é liberado.

KOPPER *et al.* (2006) afirmam que na etapa de seleção estímulos vibratórios podem ajudar o usuário a confirmar um objeto para escolha, uma vez que limitações de alguns dispositivos de RV não permitem uma evidência clara do objeto selecionado. Já na etapa de anexação, o objeto identificado por um apontador na fase anterior deve ser confirmado como o alvo da seleção, podendo o usuário utilizar um botão ou um comando (gesto, voz), para indicar uma correta seleção, e um estímulo para mostrar que o comando ou o botão pressionado foram detectados pelo sistema.

A tarefa de posicionamento consiste na manipulação do objeto, colocando-o em uma posição conforme um objetivo pré-estabelecido. Nesta etapa, os estímulos táteis poderiam informar que o objeto está próximo do objetivo ou uma colisão entre objetos ocorreu. Na etapa de liberação, os estímulos podem ser úteis para indicar ao usuário que o objeto foi liberado realmente (KOPPER *et al.*, 2006).

Para auxiliar o usuário nos testes, que consistiam na montagem de um quebra-cabeça virtual, em cada etapa foram usados alternadamente estímulos táteis e visuais.

O dispositivo apresentado por KOPPER *et al.* (2006) foi implementado com uma placa de controle constituída de quatro relés de saída e pequenos motores vibratórios colocados em

partes do corpo do usuário, sendo que a interface de controle se comunicava com o computador via porta paralela, por onde recebia sinais da aplicação, indicando qual motor deveria ser ligado ou desligado, já que os quatro motores poderiam ser acionados de forma independente.

Dois tipos de testes foram realizados: um com estímulos visuais e outro com estímulos táteis. Variáveis de controle e medição foram definidas para cada etapa do processo interativo. Desta forma, nas etapas de seleção e anexação, foi adotada uma métrica objetiva, que consistia na opinião do usuário sobre a efetividade do estímulo tátil, informando se havia sentido tal estímulo e se este havia sido importante na sua tomada de decisões durante a simulação. Nas etapas de liberação e posicionamento, uma métrica objetiva similar à utilizada anteriormente foi empregada, entretanto, como no posicionamento era importante a disposição correta do objeto virtual com número baixo de colisões, foram computados o tempo integral em que o usuário mantinha um determinado objeto em estado de colisão, e a distância escalar entre o bloco e sua quadrícula. Na etapa de liberação, foi computado o intervalo de tempo entre receber um estímulo e soltar o objeto. Outras informações foram coletadas, como tempo total utilizado pelo usuário em cada parte da tarefa (com cada bloco), e tempo total de toda a tarefa.

O número de participantes envolvidos no experimento foi de 20 indivíduos, na faixa etária de 18 a 27 anos, sendo 17 homens e 3 mulheres, e 16 destros. Utilizaram óculos estereoscópicos, *mouse* e rastreador de posição (KOPPER *et al.*, 2006).

Os testes com estímulos visuais e estímulos táteis tiveram um intervalo de cinco minutos, com intuito de recuperar a fadiga muscular causada ao usuário por causa do dispositivo rastreador e possibilitar um descanso visual. Nos testes com geração de estímulos táteis, dois pequenos motores foram colocados nos usuários nas regiões acima dos joelhos. Ao

final, cada usuário preencheu um questionário que possuía questões de cunho demográfico e ligadas aos testes.

Com base nos questionários, foram obtidas informações sobre os usuários quanto ao conhecimento de RV, escolaridade e idade, sendo que 60% possuíam curso superior, todos tinham conhecimento sobre RV, e a média de idade era de aproximadamente 24 anos. No que se refere ao dispositivo, 18 usuários afirmaram ter entendido o que os estímulos táteis significaram durante as etapas, auxiliando 90% deles na conclusão da tarefa, entretanto, 40% ainda optaria pela configuração convencional se pudesse escolher, e 75% dos participantes declararam que se adaptaram de forma fácil aos atuadores táteis, que produziam as vibrações.

Nas etapas de seleção e anexação de blocos no quebra-cabeça virtual, 70% dos indivíduos sentiram e tiveram certeza do momento que o objeto havia sido selecionado, tendo uma percepção imediata do bloco virtual selecionado com a vibração. Na etapa de posicionamento, os estímulos táteis foram eficientes para indicar colisões, com um ganho de 25% no tempo médio usado em colisões, sendo que 17 usuários realizaram a tarefa mais rapidamente, com 5 segundos em média, em comparação com os tempos de execução sem a presença do tato. No entanto, todos os participantes enfrentaram grandes dificuldades para posicionar os blocos nos locais exatos, devido a interferências magnéticas no dispositivo de rastreamento e confusão na percepção de profundidade no AV. Um problema encontrado foi com relação aos óculos de RV que, em alguns momentos geravam imagens em tons de cinza, prejudicando a visualização do ambiente, e conseqüentemente, o desempenho dos usuários. Na etapa de liberação foi levado em consideração o tempo entre receber um estímulo tátil e liberar o objeto virtual, que foi de 8 segundos em média, o que no mundo real seria um valor muito alto, que pode ter sido causado por interferências no dispositivo de rastreamento de posição (KOPPER *et al.*, 2006).



### 3.5 Considerações Finais

A avaliação de interação em AVs é uma tarefa complexa, pois exige que se leve em consideração o tipo de dispositivo ou dispositivos empregados, envolvendo resolução, número de graus de liberdade, campo de visão; a tarefa a ser realizada no ambiente (seleção de um objeto entre diversos, movimentação de objetos em um eixo); o perfil do usuário (experiência com dispositivos ou com relação a tarefa), e fatores que podem influenciar seu desempenho (*feedback* visual ou tátil); desempenho do computador e da aplicação na geração de respostas e *frames* por segundo; e possíveis combinações entre esses itens, como tipo do dispositivo e a tarefa a ser executada.

O objetivo é definir fatores relacionados a aspectos computacionais (tempo de resposta por exemplo), e humanos (conforto, facilidade de uso, intuição, tempo de execução, erros ocorridos), para um aperfeiçoamento ou melhor desenvolvimento de aplicações de RV. Em ferramentas voltadas ao treinamento médico, verifica-se que alguns fatores são objetivos, podendo ser medidos na própria aplicação, e outros subjetivos, necessitando de questionários. Profissionais da área devem ser consultados para auxiliar na definição de parâmetros importantes no procedimento a ser realizado.

Todos esses fatores foram considerados na implementação do presente projeto, cuja metodologia é apresentada no capítulo a seguir.

## **CAPÍTULO 4 – IMPLEMENTAÇÃO E AVALIAÇÃO DE INTERAÇÃO NO VIMET**

Muitos são os benefícios propiciados pela RV, principalmente no que diz respeito ao desenvolvimento de interfaces avançadas, que fazem com que o usuário muitas vezes se sinta dentro de um ambiente, podendo explorar o local, selecionar e manipular objetos de tal forma que não poderia realizar no mundo real, como por exemplo, viajar por dentro do corpo humano. Neste contexto, também existem aplicações nas quais o usuário não está totalmente imersivo, nas quais equipamentos como *mouse*, teclado e tela de vídeo são usados (PINHO, 2000).

De uma forma ou de outra, é imperativo que o usuário de aplicações de RV possa usar a tecnologia para aumentar seu conhecimento ou experiência dentro de uma determinada área do conhecimento. Médicos ou estudantes de Medicina, por exemplo, podem adquirir habilidades treinando procedimentos em simuladores, já que, muitas vezes, estes usuários precisam treinar em animais ou cadáveres, o que apresenta algumas desvantagens.

Para atingir um grau elevado de realismo durante uma interação, múltiplos recursos são utilizados na implementação de AVs, como imagens tridimensionais exibidas em tempo real, dispositivos especiais de entrada e saída, tais como capacetes, óculos estereoscópicos, luvas de dados e equipamentos hápticos. Estes últimos podem fornecer retorno de força e sensação tátil, semelhante a um instrumento cirúrgico, um endoscópio ou uma agulha durante um procedimento de punção, por exemplo. Além disso, um sistema de RV pode oferecer múltiplos níveis de dificuldade e situações de um procedimento, aumentando o conhecimento e a habilidade do usuário.

Diversos dispositivos podem ser usados para interação em AVs: os não convencionais, citados no parágrafo anterior, e os convencionais, como *mouse* e teclado. A seleção desses equipamentos depende de fatores como recursos disponíveis, tarefas a serem realizadas e objetivos a serem alcançados. Os dispositivos não convencionais geralmente possuem um custo elevado; no entanto, podem proporcionar um grau de realismo maior do que os comuns em determinadas tarefas.

Para facilitar a construção de aplicações de RV, incluindo suporte a múltiplos dispositivos (*mouse*, teclado, luvas, entre outros), um *framework* pode ser de grande utilidade, uma vez que pode oferecer funcionalidades necessárias e comuns a diversas aplicações, como formas diversas de interação, deformação, detecção de colisão, estereoscopia, entre outras. No Laboratório de Aplicações de Informática em Saúde (LApIS), do Centro Universitário Eurípides de Marília, está sendo implementado um *framework* para treinamento médico, o ViMeT (*Virtual Medical Training*) (OLIVEIRA, 2007), o qual não possuía suporte para interação com dispositivos hápticos e luvas de dados.

O domínio do ViMeT é constituído por aplicações que simulam exames de biópsia, procedimentos para coletar amostras de células de partes do corpo humano, visando ao encaminhamento para análise laboratorial a fim de verificar a existência de anomalias.

No presente trabalho, foi inicialmente considerado como estudo de caso um procedimento de biópsia mamária para o treinamento médico virtual. Neste caso, a agulha é o instrumento médico utilizado no exame, manuseada pelo profissional de Medicina, e é inserida no órgão (mama), até atingir um determinado alvo (nódulo), de onde se aspira o líquido de um cisto ou um pequeno número de tecido mamário (FREITAS JR., 2001). Para simular as tarefas do procedimento de biópsia, a simulação para o treinamento deve prever todas as ações necessárias desde o início do procedimento até a sua finalização, bem como

permitir a visualização do treinamento considerando diversos pontos de vista. Em um AV, cada uma dessas fases merece especial atenção em relação à interação do usuário com a aplicação.

A seguir é apresentada uma descrição do ViMeT e a implementação do módulo de interação para este *framework*. Desta forma, é possível gerar aplicações que permitam simular procedimentos de biópsia, minimizando-se a preocupação com a implementação das tarefas de interação, usando equipamentos convencionais e não convencionais.

#### **4.1 ViMeT – *Virtual Medical Training***

O ViMeT é um *framework* orientado a objetos, em desenvolvimento, voltado a disponibilizar, de forma genérica, determinadas funcionalidades e características, que foram classificadas como importantes para simulações que visem a treinar profissionais da área médica, a citar: detecção de colisão, deformação, interação com equipamentos convencionais e não convencionais, interface gráfica, modelagem de objetos tridimensionais e AVs. Como mencionado, tais funcionalidades foram projetadas para serem utilizadas na construção de aplicações para treinamento médico, inicialmente em simulações que envolvam exames de biópsia (OLIVEIRA, 2007).

Este *framework* está sendo construído em linguagem Java, usando a API Java 3D, sendo que as funcionalidades citadas são implementadas como classes, conforme pode ser observado na Figura 23. As classes são responsáveis por criar o AV (Classe *Environment*); definir parâmetros e carregar objetos tridimensionais (Classes *DefObj* – objeto que sofre deformação, *ObjRig* – objeto rígido que causa a deformação e Classe *3DObject*); determinar parâmetros do ambiente como iluminação (Classe *Light*), fornecer facilidade de estereoscopia (Classe *Steroscropy*), deformação (Classe *Deformation*) e detecção de colisão (Classe

*Collision*) e manipular banco de dados (Classes *DatabaseInterface* e *DatabaseManipulation*). As classes citadas estão envolvidas por um quadro em linha contínua, como apresenta a Figura 23. As classes apresentadas em cinza são as responsáveis pela interação, constituindo o módulo implementado no presente trabalho (*Glove*, *Haptic* e *Devices*).

As classes *Octree* (KERA, 2005), *Anaglyph* (BOTEGA, 2005), e *MassSpring* (PAVARINI, 2006), esta última juntamente com as classes *Parameters* e *Neighbors*, envolvidas na Figura 23 por quadros em negrito, foram implementadas em projetos desenvolvidos anteriormente e integrados ao ViMeT. Suas funcionalidades são implementar as funcionalidades de detecção de colisão entre os objetos virtuais, esteresocopia usando anaglifos e deformação de um objeto usando o método massa-mola, respectivamente.

Uma ferramenta de apoio, denominada *ViMeTWizard*, fornece facilidade para instanciação das classes. É possível também utilizar o *framework* sem esta ferramenta, por meio da instanciação manual de suas classes (NUNES *et al.*, 2007b).

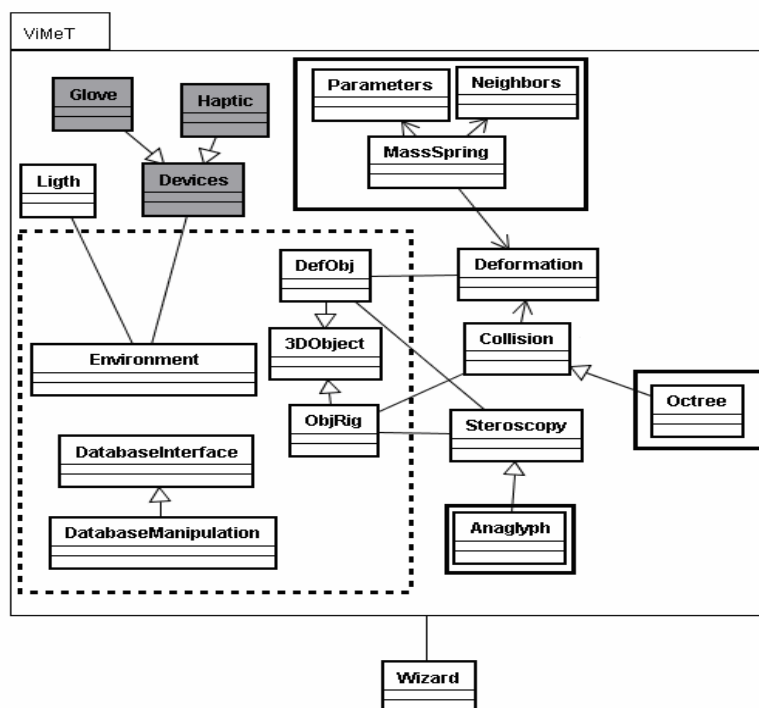


Figura 23 - Diagrama com as classes do ViMeT (NUNES *et al.*, 2007b)

Na Figura 24 é apresentado o projeto de arquitetura do ViMeT, com as formas de instanciação (diretamente pelas classes ou pela ferramenta *Wizard*), as classes responsáveis pelas funcionalidades, a camada de persistência que permite uma flexibilidade na manutenção do banco de dados que, por sua vez, armazena objetos modelados e dados das aplicações instanciadas a partir deste *framework*.

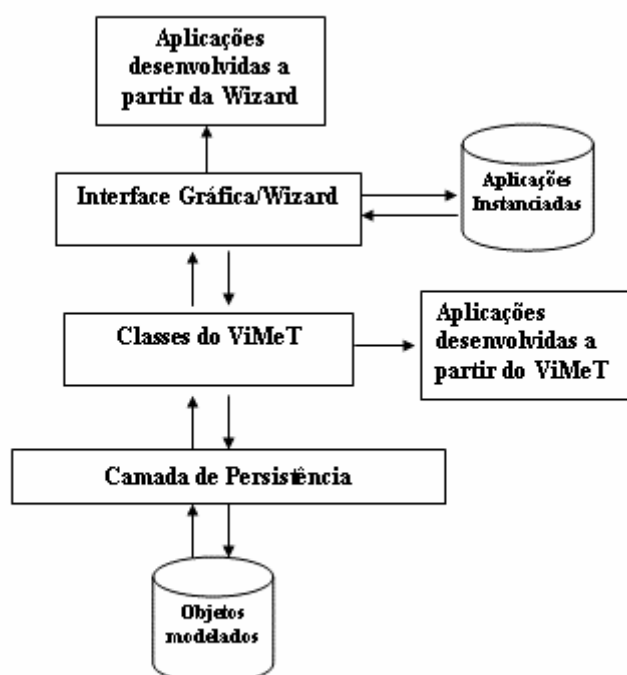


Figura 24 - Projeto da arquitetura do ViMeT (OLIVEIRA, 2007)

Inicialmente, o ViMeT foi idealizado para ser empregado na construção de aplicações que enfocam simulação de exames de punção, havendo, neste caso, a necessidade de inserir dois objetos modelados: um que representa o órgão humano sobre o qual o exame será realizado, e outro que representa o instrumento que coleta material deste órgão. Assim, o desenvolvedor seleciona os objetos e determina características no que diz respeito à colisão, deformação, estereoscopia e interação. O sistema cria um AV de acordo com as opções

escolhidas, gerando o código-fonte, que poderá ser alterado conforme as necessidades da aplicação em desenvolvimento (OLIVEIRA, 2007).

A Figura 25 apresenta a interface de uma aplicação gerada com o ViMeT antes da construção do módulo de interação, com a importação de dois objetos: um deformável, que representa o órgão humano (no caso, uma mama) e outro que representa o instrumento médico (no caso, uma seringa). Nesta aplicação foram usadas as funcionalidades de estereoscopia (anaglifos), deformação com o método massa-mola e detecção de colisão com o método *Octree*. Nesse exemplo, o *mouse* foi definido como dispositivo para manipular o instrumento médico. Uma classe construída no projeto denominada *Mouse*, permite a rotação e a translação do objeto virtual correspondente a este instrumento.

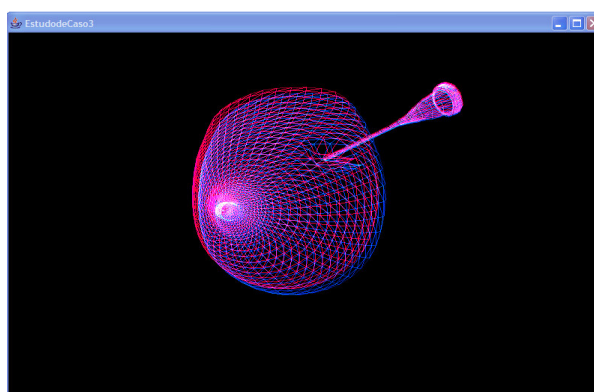


Figura 25 – Exemplo de aplicação gerada com o ViMeT (OLIVEIRA, 2007)

Outros exemplos de aplicações geradas são apresentados na Figuras 26 e 27, sem estereoscopia e com a importação de outros objetos virtuais ao invés da mama virtual.

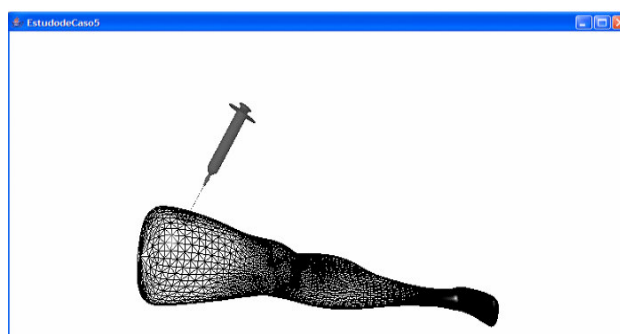


Figura 26 – Exemplo de aplicação gerada sem estereoscopia (OLIVEIRA, 2007)

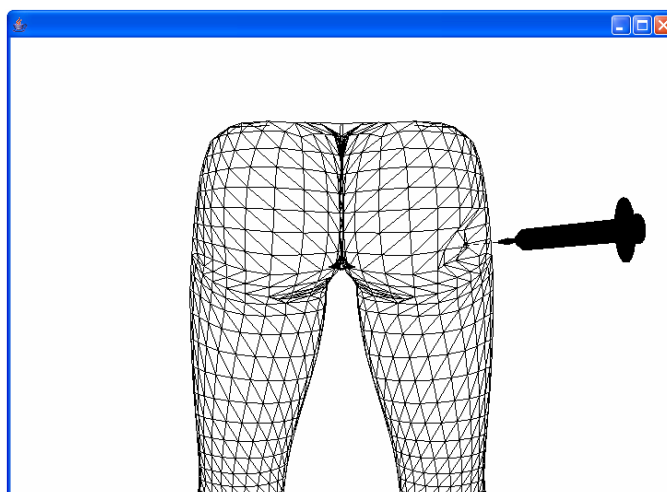


Figura 27 – Exemplo de aplicação gerada com outro objeto virtual deformável (OLIVEIRA, 2007)

## 4.2 Dispositivos do Módulo de Interação

Conforme introduzido anteriormente, o módulo de interação deve oferecer suporte a dispositivos convencionais e não convencionais. Os dispositivos convencionais adotados inicialmente são um *mouse* comum e um teclado, e os não convencionais são uma luva de dados e um equipamento háptico. A Tabela 8 apresenta os dispositivos do módulo de interação e as operações que eles podem desempenhar, com relação à navegação, seleção e manipulação, e controle do sistema. Pode-se afirmar que a seleção não ocorre em tempo de execução, mas durante a geração ou a construção da aplicação, momento em que é definido o dispositivo e o objeto virtual a ser manipulado pelo mesmo, realizando uma associação entre ambos (instrumento médico virtual e *mouse* ou instrumento médico virtual e dispositivo háptico, por exemplo).

O controle do sistema, ainda que simples até o presente momento, é realizado automaticamente, visto que as operações de navegação e manipulação podem ser realizadas sem a interrupção da interação. A seguir, são descritas as operações de navegação e manipulação implementadas e os dispositivos envolvidos em cada operação.



Tabela 8 – Dispositivos não convencionais e operações

<b>Operação</b> <b>Dispositivos</b>	<b>Navegação</b>	<b>Seleção/Manipulação</b>	<b>Controle do Sistema</b>
<i>Mouse</i>		X	
Teclado	X	X	
Luva de Dados		X	
Dispositivo Háptico		X	

O *mouse* tem como função capturar as coordenadas  $x$  e  $z$  dos movimentos executados no plano pelo participante da simulação, bem como informações sobre os botões (pressionado ou liberado), e enviar ao computador, fazendo com que a aplicação movimente um determinado objeto virtual alterando a rotação e a translação deste. Assim, este dispositivo permite que o usuário manipule o objeto virtual, efetuando a translação deste no AV durante a interação.

O teclado movimenta objetos virtuais por meio de teclas específicas e permite a navegação pelo AV por meio das setas, fazendo com que o usuário possa alterar seu ponto de vista pelo ambiente.

A luva de dados utilizada no módulo de interação do ViMeT é uma *5DT Data Glove Ultra*, com 5 sensores, mostrada na Figura 28, que mede a flexão dos dedos, permitindo a captura de movimentos, que podem ser transmitidos ao PC (*Fifth Dimension Technologies*, 2007).



Figura 28 - 5DT Data Glove 5 Ultra (*Fifth Dimension Technologies*, 2007)

O dispositivo háptico utilizado no desenvolvimento do módulo de interação do ViMeT é um PHANTOM *Omni*, mostrado na Figura 29, cujas especificações estão apresentadas na Tabela 9.

Tabela 9: Especificações do Dispositivo PHANTOM *Omni* (*SensAble Technologies*, 2007)

<b>Especificações do Produto</b>	
Resolução de Posição	Aproximadamente 0,055 mm
Espaço de Trabalho	160mm de largura 120 mm de altura 70mm de diâmetro
Fricção	Menor que 0,26 N
Força Aplicável Máxima	3,3 N
Força Aplicável Contínua (24 horas)	Maior que 0,88 N
Tenacidade	Eixo x 1,26 N/mm Eixo y 2,31 N/mm Eixo z 1,02 N/mm
Peso	Aproximadamente 1,47 Kg
Temperatura de Operação	De 10 a 35 °C
Temperatura de Armazenamento	De -40 a 65 °C
Movimento	6 graus de liberdade
<i>Feedback</i> de Força	3 graus de liberdade (x, y, z)



Figura 29 - PHANTOM *Omni* (*SensAble Technologies*, 2007)

### 4.3 Considerações Sobre a Tecnologia Adotada

No que diz respeito à tecnologia, em nível de *hardware* foram utilizados computadores pessoais, sendo o primeiro com processador *Pentium 4*, de 3.0 GHz, 1GB de RAM, com

placa de vídeo NVIDIA *GeForce 5500*, e o segundo, um *notebook* DELL, com processador *Celeron 1.6 GHz*, 512 MB de RAM, placa de vídeo *Mobile Intel® 945GM Express Chipset Family*, ambos com placas denominadas *FireWire* (IEEE 1394), para permitir a conexão entre o computador e o dispositivo háptico.

Em nível de *software*, foi considerada a tecnologia inicialmente empregada para implementar o ViMeT: linguagem de programação Java e a API Java3D, escolhidas previamente devido à possibilidade de redução de custos, portabilidade, reutilização de código e facilidades para construção de AVs, possibilitando uma maior produtividade na construção de aplicações dentro do domínio considerado com um baixo custo.

A linguagem Java foi desenvolvida na década de 90, com o intuito de ser orientada a objetos, portátil e com recursos de rede e segurança, sendo que os programas escritos em Java são compilados para *bytecodes*, que são executados por uma máquina virtual, denominada JVM (*Java Virtual Machine*). Além disso, oferece sintaxe similar às linguagens de programação C e C++, possui um grande número de APIs para desenvolver aplicações em diversas áreas, inclusive RV, carga dinâmica de código e desalocação automática de memória, esta última realizada pelo *garbage collector*, facilidade para implementação de aplicações distribuídas e multitarefa (DEITEL; DEITEL; 2001).

A API Java3D é uma biblioteca com diversas classes e métodos que facilitam a criação e especificação de comportamento de AVs, trabalhando por meio de uma estrutura hierárquica de nós denominada grafo de cena, que define as propriedades (geometria, textura, cor) e as relações entre os objetos de uma cena. A Figura 30 apresenta a estrutura simplificada de um grafo de cena em Java3D e a representação dos nós comumente utilizados (nó de definição do universo virtual, nós de localização no universo, nós de grupo para indicar relacionamento

com outros nós ou grupo de nós e transformações, como por exemplo, orientação e translação de um determinado objeto) (JAVA3D, 2008).

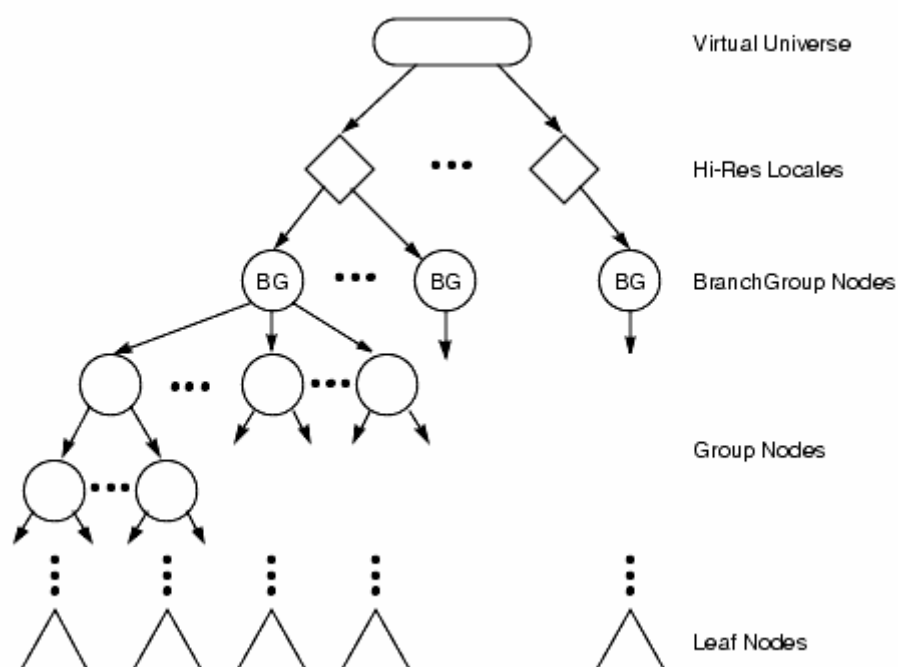


Figura 30 – Estrutura do Grafo de Cena em Java3D

Esta API também permite a importação de modelos tridimensionais construídos em outros aplicativos, além da facilidade para inclusão de determinados dispositivos de RV e definição de som em 3D.

Entretanto, um problema foi encontrado nesta etapa do projeto: o ViMeT está sendo implementado em Java e os dispositivos não convencionais (luva de dados e dispositivo háptico), possuem *drivers* e bibliotecas em linguagem C, disponibilizados pelos fabricantes. Estas bibliotecas têm funções pré-definidas, que facilitam o desenvolvimento de aplicações que utilizam tais equipamentos.

Dessa forma, duas abordagens poderiam ser adotadas: a construção de *drivers* próprios, em linguagem Java, ou a integração das diferentes linguagens de programação,

possibilitando uma interface entre a aplicação em Java e as bibliotecas e *drivers* em linguagem de programação C.

A primeira apresentaria um desvio no propósito do presente projeto devido a questões, como: programação em baixo nível, abrangência de diversos tipos de portas de entrada para conexão dos equipamentos, tempo dedicado à implementação e testes, que poderia ser utilizado no aperfeiçoamento da aplicação.

A segunda abordagem eliminaria algumas dessas questões, pois os *drivers* e funções já estão testados e aprovados. Entretanto, exige o conhecimento das linguagens adotadas e pesquisa intensa sobre as funcionalidades de ambas, bem como o estudo das bibliotecas disponíveis.

A abordagem adotada foi a segunda, visto que também seria possível oferecer uma contribuição adicional em relação à disponibilização de material técnico detalhado sobre integração de linguagens, escasso na literatura. Para a realização da integração entre as linguagens de programação C/C++ e Java, foi utilizada uma interface de programação nativa, denominada JNI (*Java Native Interface*), que permite a interoperação entre a aplicação escrita em Java e bibliotecas e programas desenvolvidos em outras linguagens, como C, C++ e *Assembly* (CORNELL; HORSTMANN; 2003).

Para tornar a integração possível, o JDK (*Java Development Kit*) oferece bibliotecas, como *jni.h* e *jni\_md.h*, e ferramentas, como *javah*, que permitem a construção de métodos nativos, a compilação dos programas em Java e a geração de arquivos de cabeçalho para a criação de bibliotecas de ligação, o carregamento de tais bibliotecas, a especificação de chamadas e ligações entre os métodos JNI (Java) e as funções nativas (Linguagem C) (SUN, 2007).

No presente trabalho, a plataforma utilizada foi *Windows XP*; no entanto, outros Sistemas Operacionais podem ser utilizados. A empresa *SensAble Technologies*, por exemplo, fabricante do dispositivo háptico *PHANTOM Omni*, oferece *drivers* e bibliotecas para *Linux* e *MacOS*.

O compilador escolhido para gerar a biblioteca de ligação foi o aplicativo *Microsoft Visual C++ 6.0*, devido à familiaridade com esta tecnologia. Entretanto, outros compiladores podem ser adotados, como *Microsoft Visual C++ Toolkit 2003*, *Borland C++ Compiler*, *Bloodshed Dev-C++* (CORRÊA *et al.*, 2008).

O *Microsoft Visual C++ 6.0* consiste em um IDE (*Integrated Development Environment*) ou Ambiente de Desenvolvimento Integrado, desenvolvido pela *Microsoft* para as linguagens de programação C, C++. É constituído de ferramentas para facilitar o desenvolvimento de aplicações voltadas especialmente para as APIs do ambiente *Windows*, bem como a API *DirectX* e o *framework Microsoft.NET* (Microsoft, 2008). O Apêndice B descreve detalhes da configuração do *Microsoft Visual C++ 6.0* para permitir a geração da biblioteca de ligação entre a aplicação em Java e as bibliotecas do dispositivo háptico.

O intuito é usar tecnologias disponibilizadas gratuitamente, o que contribui para diminuir o custo de aplicações desta natureza, uma vez que a aquisição de dispositivos não convencionais apresenta um custo elevado.

#### **4.4 Bibliotecas dos Dispositivos Não Convencionais**

Os fabricantes do dispositivo háptico *PHANTOM Omni* e da luva de dados *5DT DataGlove Ultra*, *SensAble Technologies* e *Fifth Dimension Technologies*, respectivamente, fornecem CDs com *drivers* e bibliotecas contendo funções escritas em linguagens de

programação C e C++, com a finalidade de realizar determinadas operações com tais equipamentos, permitindo o controle dos mesmos.

Com relação às funções dos dispositivos não convencionais do módulo de interação, as assinaturas do *software* da luva de dados são compostas pelo identificador *fd*, indicando o fabricante (*Fifth Dimension Technologies*); em seguida, é usado um verbo para definir uma determinada ação (*Scan, Get, Open, Close*), e a informação que identifica o propósito da função (*SensorRawAll, SensorScaledAll, NumberSensors, Calibration*). Em seguida, entre parênteses são disponibilizados os parâmetros da referida função, geralmente compostos pelo ponteiro do tipo *fdGlove* (que representa a luva de dados na aplicação e é definido na biblioteca disponibilizada pelo fabricante) e uma variável para armazenar informações a serem recebidas (como os valores dos sensores), ou a serem enviadas (como por exemplo, valores para calibração).

Por exemplo, na função *fdGetSensorRawAll(pGlove, vetor)*, que pode ser implementada para obter os valores de todos os sensores da luva de dados simultaneamente, *fd* indica o fabricante, *Get* identifica a ação de receber informação, *SensorRawAll* mostra que é uma função para obtenção de todos os sensores do dispositivo, *pGlove* é um ponteiro do tipo *fdGlove*, e *vetor* indica um *array* fictício para armazenar os valores obtidos.

O *OpenHaptics Toolkit*, fornecido pelo fabricante, permite a integração com aplicações usando OpenGL, podendo especificar geometrias e suplementá-las com comandos de simulação de propriedades hápticas (fricção e tenacidade). Suporta diversas plataformas (*Linux, Windows XP e 2000, Macintosh*) e inclui as APIs HDAPI (*Haptic Device API*) e HLAPI (*Haptic Library API*), utilitários, códigos de exemplos e PDD (*PHANTOM Device Drivers*). A HDAPI oferece acesso ao dispositivo em baixo nível, permitindo a renderização direta de forças e controle dos *drivers*. Já a HLAPI oferece um alto nível de renderização

háptica, facilitando inclusive a integração com o *OpenGL*, conforme apresentado na Figura 31.

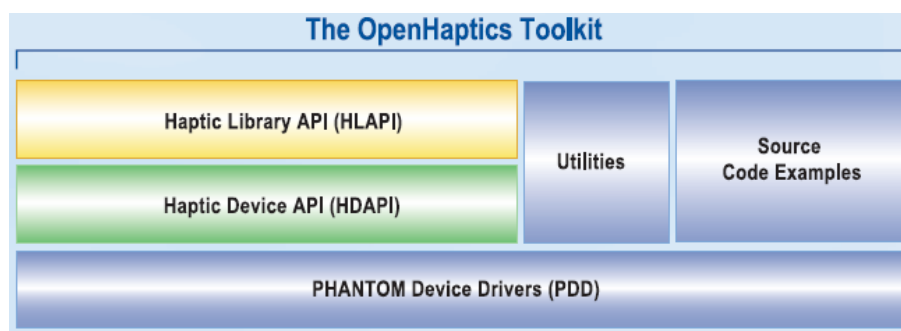


Figura 31 - Diagrama de camadas do *OpenHaptics Toolkit* (SensAble Technologies, 2007)

As funções do *OpenHaptics Toolkit* são formadas inicialmente pela identificação da biblioteca a ser utilizada, *hd* para HDAPI e *hl* para HLAPI, seguida de um verbo para indicar uma ação a ser realizada (*Get, Set, Begin, End*) e o tipo de dado ou entidade (*Double, Integer, Device, Frame*). Entre parênteses, geralmente em letras maiúsculas e também com a identificação da biblioteca (HL ou HD), deve-se fornecer o complemento da ação a ser realizada, que é constituído por variáveis, que podem ser empregadas para receber ou enviar informações para o dispositivo, e constantes, tais como: HD\_CURRENT\_POSITION, para determinar a posição do braço do equipamento nos três eixos; HD\_FORCE\_OUTPUT, para indicar a habilitação do retorno de força; HD\_CURRENT\_TRANSFORM, para indicar a obtenção de um vetor de 16 posições para formar uma matriz de transformação com 4 linhas e 4 colunas, permitindo a identificação dos movimentos de rotação nos três eixos do braço do dispositivo.

O diagrama apresentado na Figura 32 mostra a lógica de um programa construído com a camada HDAPI do *OpenHaptics Toolkit* para a renderização de objetos virtuais, com as funções a serem utilizadas para inicializar e finalizar o dispositivo, habilitar e desabilitar funções, calcular o retorno de força, receber e enviar informações.



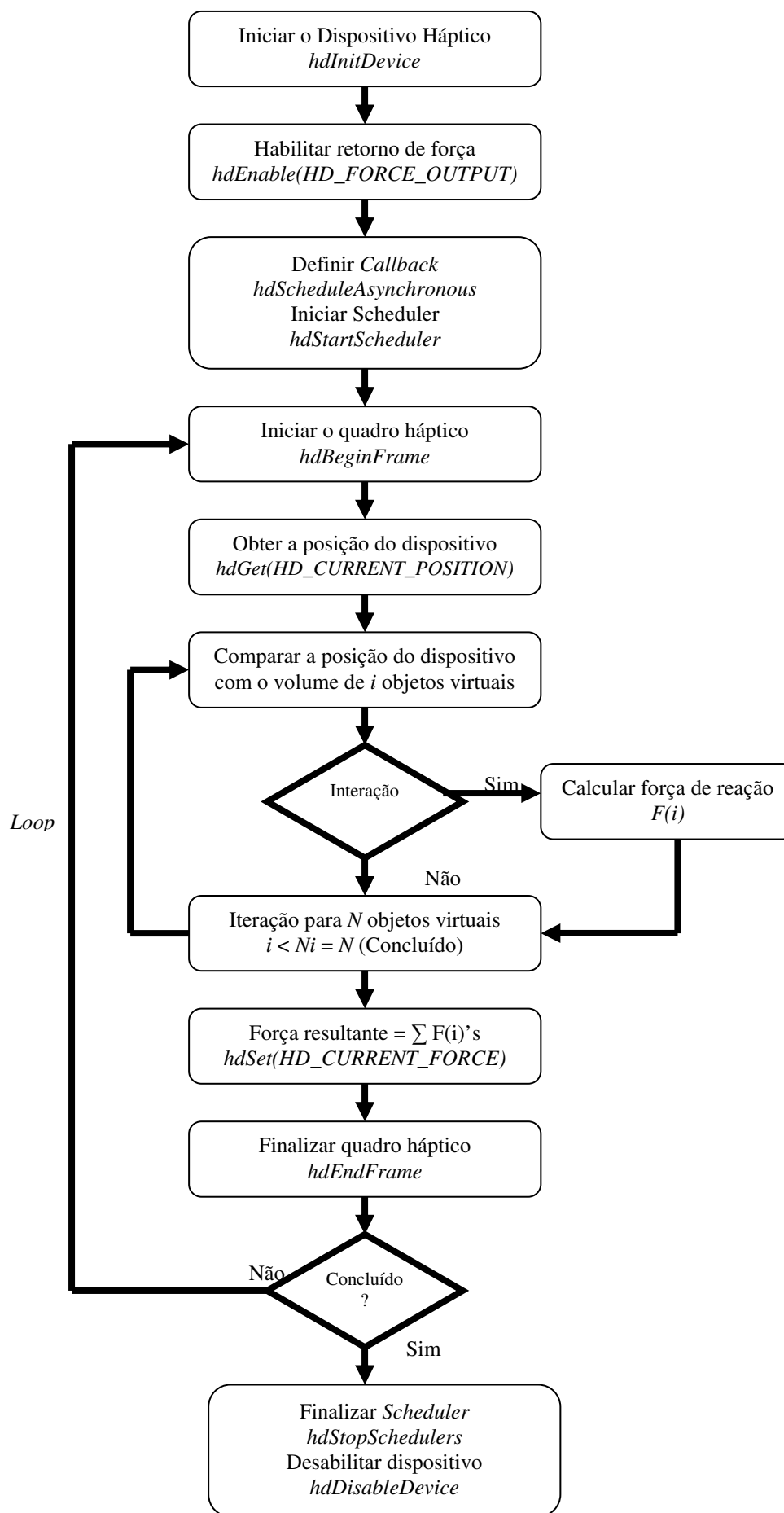


Figura 32 – Diagrama de um Programa HD-API (SensAble Technologies, 2007)

Por exemplo, na função *hdGetDoublev(HD\_CURRENT\_POSITION, position)*, *hd* indica que é uma função de baixo nível (HDAPI), *Get* indica que uma informação será requerida e *Doublev* é o tipo da informação. Entre parênteses estão os parâmetros que formam o complemento, no caso, *HD\_CURRENT\_POSITION*, para recebimento da posição atual do equipamento (translação nos eixos x, y e z) via HDAPI (nomenclatura do parâmetro iniciada com HD), e *position*, um vetor de três posições do tipo *double* para armazenar os dados.

O diagrama da Figura 33 apresenta, por sua vez, a estrutura de um programa HLAPI, que utiliza funções do OpenGL, realizando as renderizações háptica e gráfica, com o mapeamento das coordenadas hápticas para o AV.

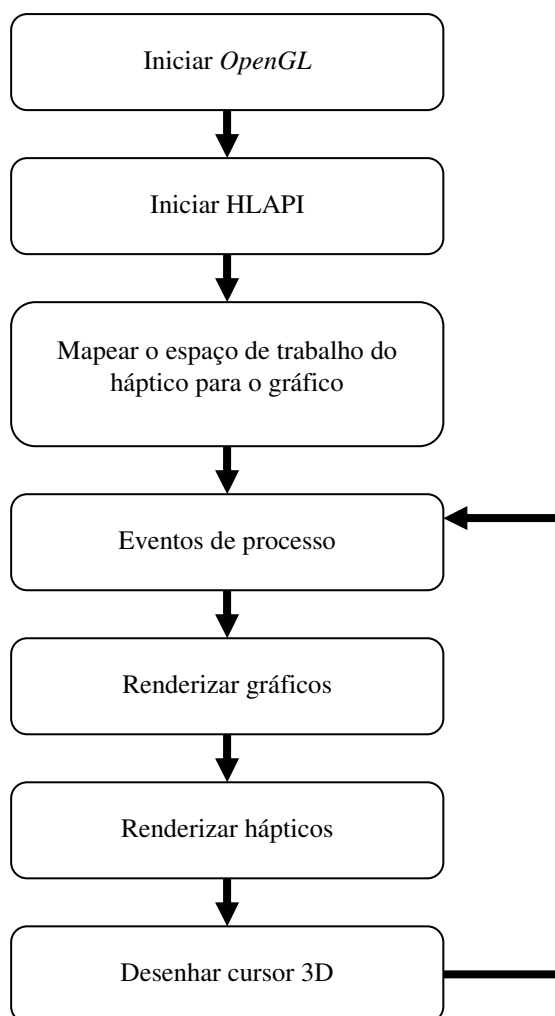


Figura 33 – Diagrama de um Programa HLAPI (*SensAble Technologies*, 2007)

## 4.5 Visão Geral do Módulo de Interação

Como mencionado, o módulo de interação é composto por dispositivos convencionais (teclado e *mouse* comum) e não convencionais (dispositivo háptico e luva de dados) para propiciar a comunicação entre usuário e computador.

A Figura 34 mostra a dinâmica do módulo de interação, com os dispositivos associados aos seus respectivos objetos virtuais, sendo que as setas vermelhas indicam o fluxo de informações, podendo se observar que entre o equipamento háptico e o instrumento médico virtual há uma seta bidirecional, que contempla além do recebimento de informações, o retorno de força.

Pode-se notar, ainda, as combinações de dispositivos permitidas ao desenvolvedor: *mouse* e teclado, *mouse* e luva de dados, equipamento háptico e teclado, equipamento háptico e luva de dados, mesclando, desta maneira, a utilização de equipamentos convencionais e não convencionais. O desenvolvedor pode optar também pela utilização de um dispositivo por vez em cada sessão de treinamento.



Figura 34 - Funcionamento do módulo de interação

Com relação ao projeto e desenvolvimento do módulo para funcionamento dos diversos equipamentos, algumas questões foram levadas em consideração. Como no ViMeT

classes e métodos implementados em Java são responsáveis pelas funcionalidades e características, o módulo de interação para a comunicação entre o sistema e dispositivos (*mouse*, teclado, luvas e equipamento háptico), foi construído da mesma forma, incluindo as classes *Glove*, *Haptic*, *Keyboard*, *NativeGlove*, *NativeHaptic* no projeto, conforme o diagrama apresentado na Figura 35. A classe *Mouse* já estava implementada para propiciar a interação por meio do dispositivo *mouse* gerenciando a manipulação do objeto que representa o instrumento médico.

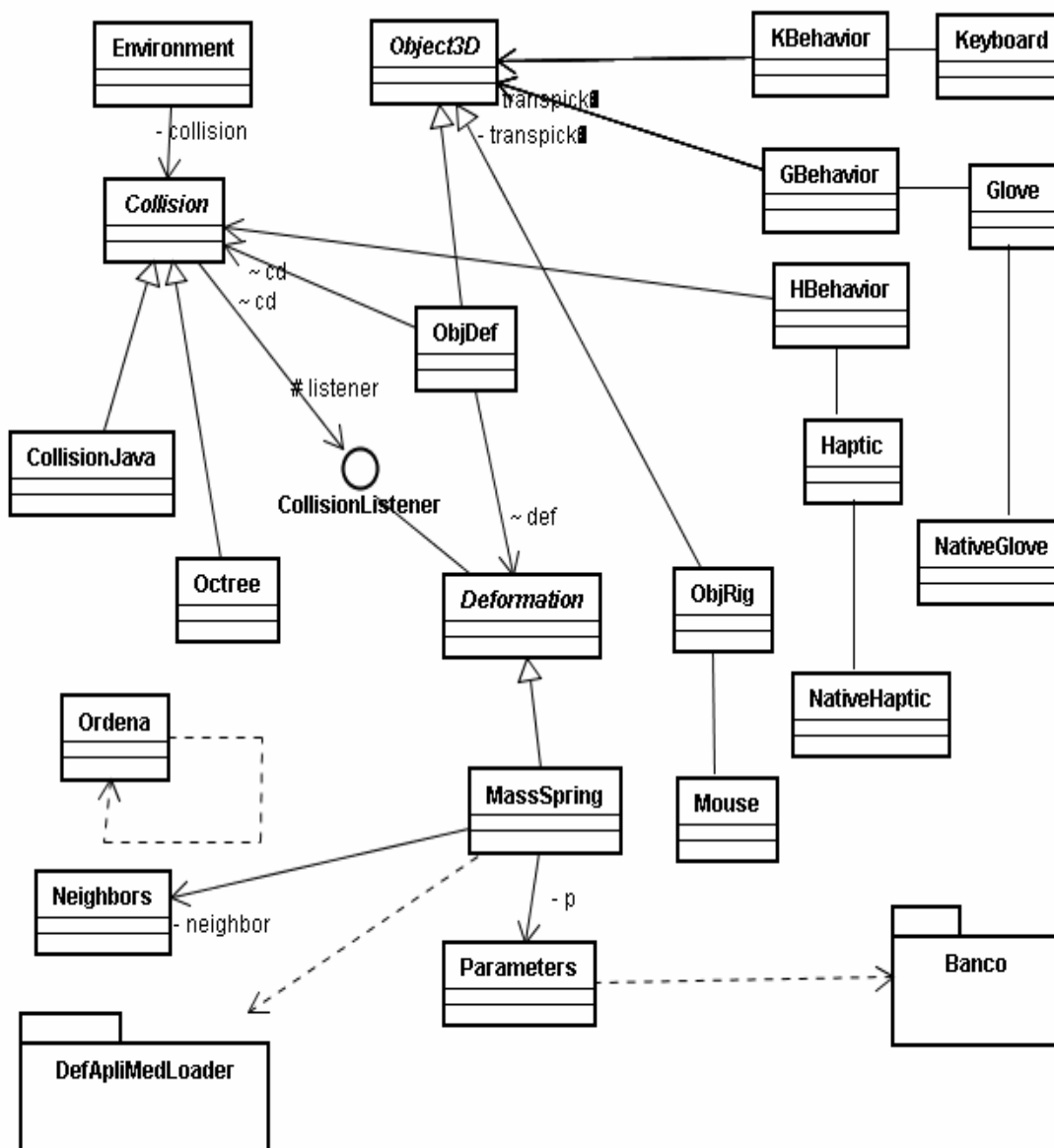


Figura 35 – Diagrama de classes atualizado do ViMeT

Observa-se que a classe abstrata *Devices*, proposta anteriormente (Figura 23) não foi implementada, e com as classes *Haptic* e *Glove* foram estabelecidos relacionamentos com a classe *Object3D* por meio das classes *Behaviors* (*HBehavior* e *GBehavior*), semelhante à classe *Keyboard* (*KBehavior*), não havendo o relacionamento direto com a classe *Environment*, conforme proposta inicial de implementação. Isso ocorreu porque os dispositivos foram associados aos objetos virtuais para manipulação dos mesmos. A navegação usando as setas do teclado foi implementada com métodos da Java3D, inseridos na classe *Environment*, conforme o trecho de código apresentada na Figura 36.

```
KeyNavigatorBehavior keyNav = new KeyNavigatorBehavior (viewXfmGroup);
keyNav.setSchedulingBounds (movingBounds);
viewBranch.addChild (keyNav);
```

Figura 36 – Trecho do código de navegação

A classe *Keyboard* possui os métodos disponibilizados pela linguagem de programação Java, implementados usando a interface *KeyListener*, conforme é mostrado na Tabela 10.

Tabela 10: Principais métodos da classe *Keyboard*

<b>Métodos</b>	<b>Função</b>
<code>KeyPressed ( )</code>	Indica o pressionamento de uma determinada tecla
<code>KeyReleased ( )</code>	Indica a liberação de uma determinada tecla
<code>KeyTyped ( )</code>	Indica o pressionamento de uma tecla, a qual não é uma tecla de ação
<code>getKeyChar ( )</code>	Obtém o nome de uma tecla

As classes *NativeGlove* e *NativeHaptic* possuem os métodos responsáveis pela execução das funções na linguagem C++, para a luva de dados e o equipamento háptico, respectivamente, conforme mostram as Tabelas 11 e 12.

Tabela 11: Principais métodos da classe *NativeGlove*

<b>Métodos</b>	<b>Função</b>
openGlove( )	Abre conexão com a luva, verifica portas existentes e o número de sensores
closeGlove ( )	Fecha a comunicação com a luva de dados
getRawSensorData( )	Obtém os valores de todos os sensores
getScaledSensorData ( )	Obtém as escalas de todos os sensores
setCalibration(upper, lower)	Realiza a calibração em um intervalo de dois valores
getGesture ( )	Obtém o número do gesto
getSensorRaw (number)	Obtém o valor de um único sensor
getSensorScaled (number)	Obtém a escala de um único sensor

Tabela 12: Principais métodos da classe *NativeHaptic*

<b>Métodos</b>	<b>Função</b>
initializeDevice ( )	Inicializa o dispositivo
verifyCallback ( )	Verifica a execução da <i>callback</i>
finishDevice ( )	Finaliza o dispositivo
sendData (dataarray)	Envia dados usando um vetor, como o retorno de força nos eixos x, y e z
receiveData ( )	Recebe dados, como uma matriz de transformação 4x4
receiveTranslation ( )	Recebe a translação, valores nos eixos x, y e z
Button1State ( )	Verifica se o botão 1 está pressionado ou não
Button2State ( )	Verifica se o botão 2 está pressionado ou não
getStiffness ( )	Obtém a tenacidade máxima do equipamento

O método *finalize*, disponibilizado na linguagem de programação Java e presente em ambas as tabelas, é empregado para fechar a conexão com a luva de dados. É usado, ainda, para interromper o dispositivo háptico e suas funções de *callbacks*, antes que o *garbage collector* inicie a desalocação de memória.

As classes *Glove* e *Haptic* realizam as transformações nos objetos virtuais, utilizando os métodos nativos das classes *NativeGlove* e *NativeHaptic*. Essas duas classes, por sua vez, bem como a classe *Keyboard*, estão ligadas aos objetos que serão manipulados pela luva de dados, dispositivo háptico e teclado, respectivamente, e possuem uma classe estendida da classe *Behavior*, disponibilizada pela API Java3D, para propiciar (por meio da classe *WakeupOnElapsedFrames* e dos métodos *initialize* e *processStimulus*), a geração de um *loop* que verifica informações dos dispositivos e mapeia-as para os objetos virtuais correspondentes.

A classe *WakeupOnElapsedFrames* permite o controle de execução de um bloco de código, definindo que este ocorra de acordo com um número específico de *frames* ou quadros. No caso do parâmetro desta classe ser zero, a execução de tal bloco ocorrerá a cada quadro e, no caso do parâmetro receber o valor 5, a execução do bloco será a cada 5 quadros.

#### 4.5.1 Definição de Integração e Interação com Dispositivos Não Convencionais

Como mencionado, uma questão importante é a integração entre as linguagens de programação C++ e Java para funcionamento dos dispositivos não convencionais. As Figuras 37 e 38 apresentam os diagramas de integração e interação para a luva de dados e o dispositivo háptico. Ambos representam os fluxos de informações entre as partes da aplicação no computador, e entre o usuário e os dispositivos, demonstrados pelas setas.

Na Figura 37 pode-se observar as informações da luva de dados sendo captadas por funções escritas em linguagem C, de bibliotecas oferecidas pelo fabricante e transferidas à aplicação através de uma biblioteca de ligação, no caso, *5DTGlove.dll*. Tais informações são tratadas pela aplicação desenvolvida em Java e Java3D, que mostra ao usuário no monitor de vídeo, os movimentos do objeto virtual associado ao equipamento. Desta forma, quando os dedos do usuário são flexionados, os objetos virtuais que representam os dedos são rotacionados.

A parte da aplicação implementada em linguagem de programação C poderia receber informações da parte da aplicação desenvolvida em linguagem de programação Java, utilizando a biblioteca de ligação *5DTGlove.dll* (adicionando duas setas para baixo, da aplicação em Java para a biblioteca de ligação, e desta para a biblioteca do fabricante), no caso de um método nativo ser definido para a calibração do intervalo de valores da luva de dados por meio de dados definidos pelo usuário. Entretanto, tais valores não seriam

transferidos ao usuário, visto que a luva de dados não possui nenhum mecanismo para retorno de força, por exemplo.

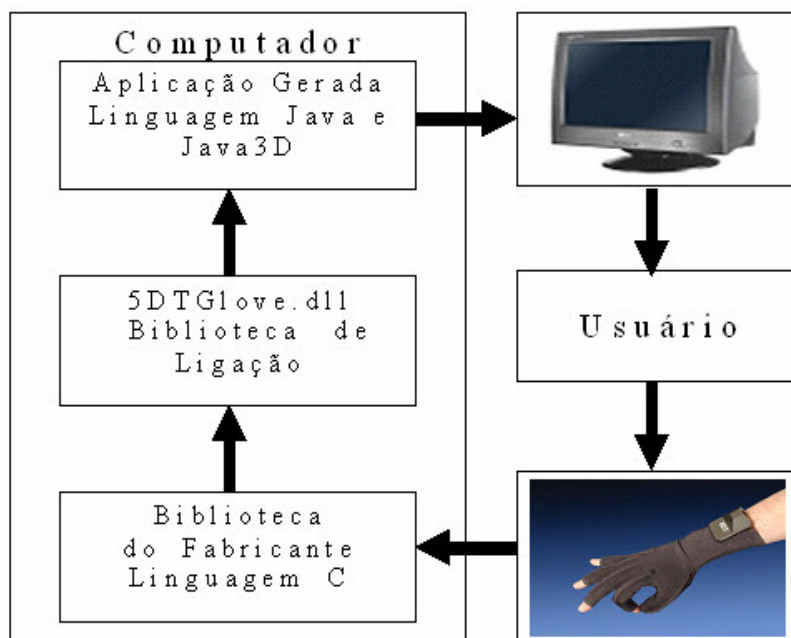


Figura 37 - Diagrama de integração e interação para a luva de dados

Na Figura 38 é apresentado um diagrama semelhante ao anterior. Entretanto, a biblioteca de ligação é denominada *Haptic.dll*, e permite a comunicação entre a aplicação desenvolvida em Java e Java3D, e as funções escritas em linguagem C.

Pode-se observar a existência de setas em ambas as direções, indicando que a aplicação, além de receber informações do dispositivo háptico, captadas pelas funções do *software OpenHaptics Toolkit*, fornecido pela *SensAble Technologies*, também processa e envia informações ao dispositivo para promover o *feedback* tátil, utilizando a biblioteca de ligação. Desta forma, o equipamento háptico se comporta como um dispositivo de entrada e saída ao mesmo tempo.



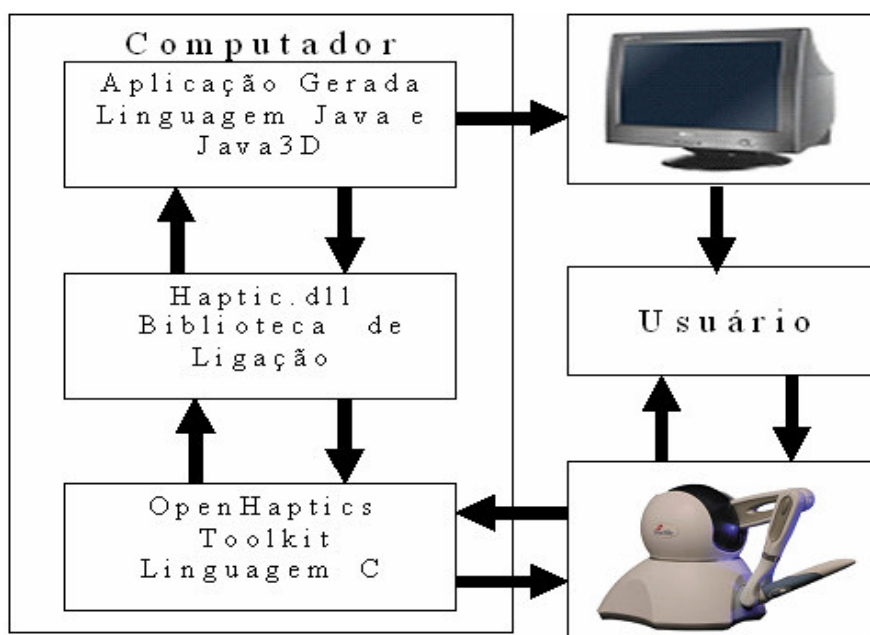


Figura 38 - Diagrama de integração e interação para o dispositivo háptico

O monitor de vídeo, presente em ambas as figuras, apresenta os objetos virtuais, suas características, ações e reações dentro do AV. Desta forma, pode-se dizer que na aplicação, há a ocorrência dos retornos visual e tátil durante a interação com o dispositivo háptico.

#### 4.6 Integração entre as Linguagens de Programação Java e C++

Como descrito anteriormente, a linguagem de programação Java disponibiliza uma série de bibliotecas e ferramentas para trabalhar com métodos nativos. Nesta seção, serão apresentadas as etapas para realizar a tarefa de integração de linguagens de programação, levando-se em consideração as linguagens C++ e Java.

Inicialmente o programa em Java deverá ser escrito. Para se trabalhar com métodos nativos, deve-se colocar a palavra-chave *native* na definição dos métodos da classe em Java, indicando à JVM que tais métodos serão executados em código nativo.

Além disso, deve ser empregado o método *System.loadLibrary*, que recebe como parâmetro a biblioteca de ligação a ser gerada. Neste caso, como o Sistema Operacional adotado foi o *Windows XP*, tais bibliotecas recebem o nome de DLLs (*Dynamic-Link Libraries*).

A Figura 39 apresenta uma classe fictícia denominada *ClasseNativa*, que contém um método nativo. Pode-se observar que a palavra-chave *native* indica que o método “NomeMetodo” será executado em outra linguagem. A palavra “ArquivoDLL”, por sua vez, representa a biblioteca de ligação a ser carregada e a nomenclatura *short[]* indica que um vetor do tipo *short* será retornado.

```
public class ClasseNativa {
    public native short[] NomeMetodo();

    static
    {
        System.loadLibrary("ArquivoDLL");
    }
}
```

Figura 39 - Código exemplo de uma classe

Em seguida, é necessário gerar um arquivo de cabeçalho com a extensão *.h*, o que pode ser realizado com o uso da ferramenta *javah* a partir do arquivo *.class*, criado na compilação da classe. No *prompt* de comandos, após a compilação do arquivo *.java* e a geração do arquivo *.class*, é necessário executar o seguinte comando no diretório onde se encontra o arquivo *.class*, para criar o arquivo de cabeçalho: *javah -jni ClasseNativa.java*.

Quando a classe pertence a um pacote, como é o caso do ViMeT, o comando com algumas alterações deve ser executado no diretório pai do diretório onde está a classe com os métodos nativos, da seguinte forma: *javah -jni -classpath . Pacote.ClasseNativa.java*.

Pode-se notar a assinatura do método em código nativo, presente no arquivo de cabeçalho (Figura 40), e no programa em C++ (Figura 42). Ambos contêm o prefixo Java, o

nome do pacote, o nome da classe, e o nome do método, separados por *underline* na definição da função nativa.

```

/*DO NOT EDIT THIS FILE - it is machine generate*/
#include <jni.h>
/*Header for class Pacote_ClasseNativa*/

#ifndef_Included_Pacote_ClasseNativa
#define_Included_Pacote_ClasseNativa
#ifdef__cplusplus
extern "C" {
#endif
/*
 *Class:    Pacote_ClasseNativa
 *Method:   NomeMetodo
 *Signature: () [S
 */
JNIEXPORT jshortArray JNICALL
Java_Pacote_ClasseNativa_NomeMetodo(JNIEnv*, jobject);
#ifdef__cplusplus
}
#endif
#endif

```

Figura 40 - Exemplo de arquivo de cabeçalho

As *strings JNIEXPORT* e *JNICALL*, definidas no arquivo *jni.h*, são usadas para especificar as chamadas e as ligações entre os métodos JNI e as funções nativas. *JNIEnv* é o ponteiro da interface JNI, que aponta para uma matriz de ponteiros e estes, por sua vez, apontam para as funções JNI. Este ponteiro é indispensável na definição de métodos nativos. O tipo *jobject* indica que um método não-estático está sendo usado, e este obtém uma referência para o objeto, como um argumento *this* implícito. O diagrama apresentado na Figura 41 mostra a hierarquia dos tipos nativos na linguagem Java.

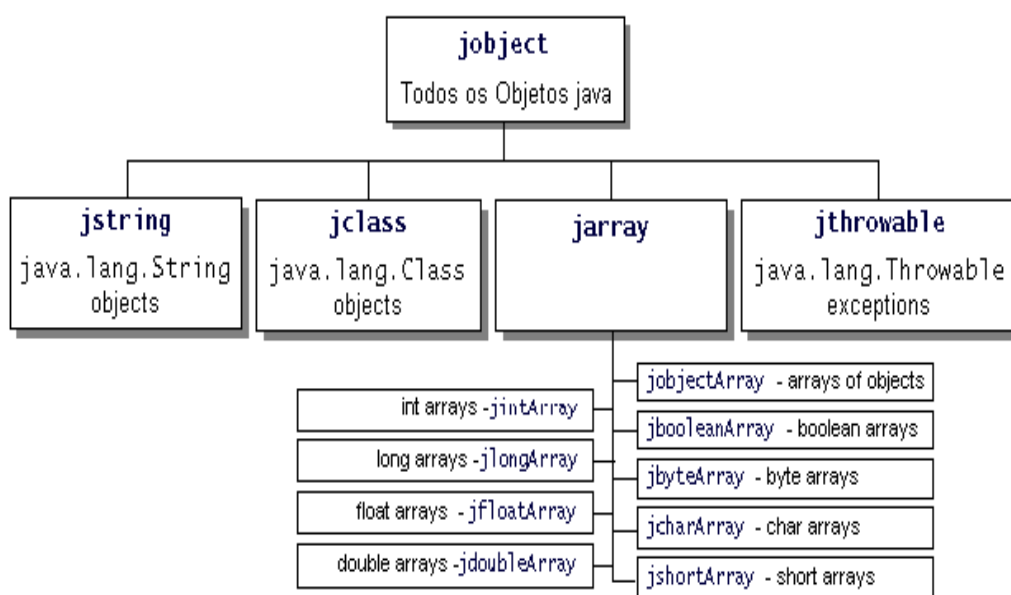


Figura 41 – Diagrama Hierárquico de tipos nativos

Há uma correspondência entre os tipos de dados e *arrays* da linguagem Java e da linguagem C para se trabalhar com métodos nativos. No exemplo, o tipo Java *short[]* corresponde ao tipo C *jshortArray*, indicando um vetor do tipo *short*. A Tabela 13 apresenta alguns tipos em Java e seus correspondentes em JNI.

Tabela 13: Tipos em Java e JNI

Tipo em Java	Tipo JNI em C
Int	Jint
Short	Jshort
Boolean	Jboolean
double	Jdouble
Float	Jfloat
int[]	jintArray

Em seguida, o programa em linguagem C é elaborado contendo as funções que serão chamadas pelos métodos escritos em Java. No cabeçalho deste programa devem ser invocados o arquivo de cabeçalho (*ClasseNativa.h*), gerado na etapa anterior, e o arquivo *jni.h*, que serão utilizados pelo compilador para gerar a biblioteca.

```

#include <jni.h>
#include "ClasseNativa.h"
...

JNIEXPORT jshortArray JNICALL
Java_Pacote_ClasseNativa_NomeMetodo
(JNIEnv *env, jobject obj)
{
    //Função para obter valores dos sensores
}
...

```

Figura 42 - Exemplo de código nativo

O último passo consiste na compilação deste programa escrito em C++ para gerar a biblioteca de ligação, que é copiada para o diretório principal do pacote do *framework* ViMeT.

## 4.7 Integração dos Dispositivos Convencionais

Os dispositivos convencionais (*mouse* e teclado) são tratados com recursos da linguagem Java e da API Java3D.

Desta forma, para a utilização do teclado no módulo de interação, foi utilizada uma *interface* disponível na linguagem Java, denominada *KeyListener*, que define os eventos desse dispositivo e permite a utilização de diversos métodos, tais como: *keyPressed*, responsável pela indicação do pressionamento de uma determinada tecla; *keyReleased*, responsável pela indicação da liberação de uma determinada tecla; *keyTyped*, responsável pela indicação do pressionamento de uma tecla que não seja de ação e *getKeyChar*, que permite o retorno de um *string* com o nome da tecla pressionada (BEZERRA *et al.*, 2008).

Para integração do *mouse* foram utilizadas classes disponíveis na API Java3D, tais como: *MouseRotate*, que verifica movimentos de rotação causados pelo dispositivo e *MouseTranslate*, que verifica movimentos de translação no eixos *x* e *y*, e *MouseZoom*, que

verifica movimentos de translação no eixo  $z$ , sendo acionadas de acordo com o botão a ser pressionado pelo usuário.

As Figuras 43 e 44 apresentam o funcionamento do teclado e do *mouse*, no qual o usuário manipula a mão virtual e o objeto que representa o instrumento médico (seringa), respectivamente, sendo que as reações são apresentadas na tela do monitor.

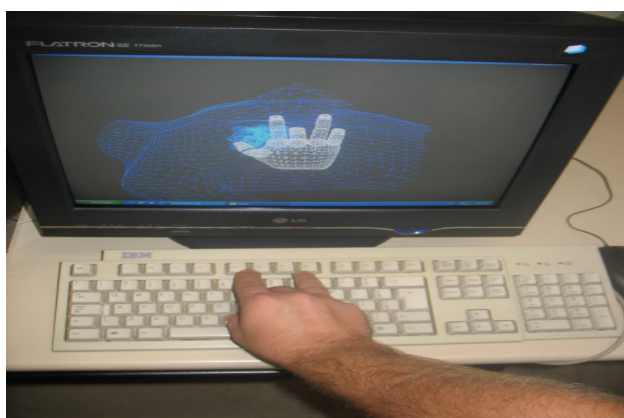


Figura 43 – Utilização do teclado

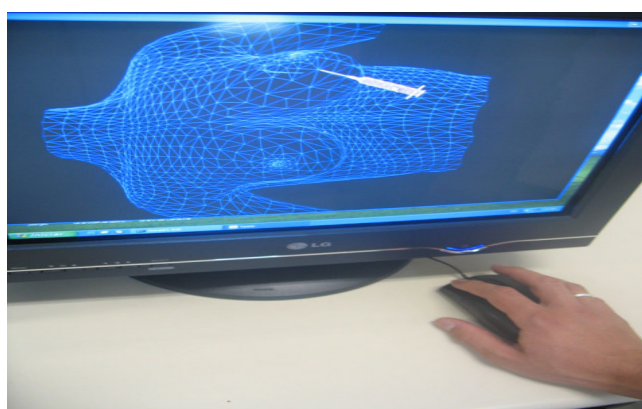


Figura 44 – Utilização do *mouse*

## 4.8 Integração do Dispositivo Háptico

O *OpenHaptics Toolkit* também permite o trabalho com mais de um dispositivo háptico simultaneamente. O diagrama da Figura 45 apresenta as etapas do código escrito em C para este projeto, no qual foram utilizadas funções HDAPI para implementações, incluídas no cabeçalho do programa.

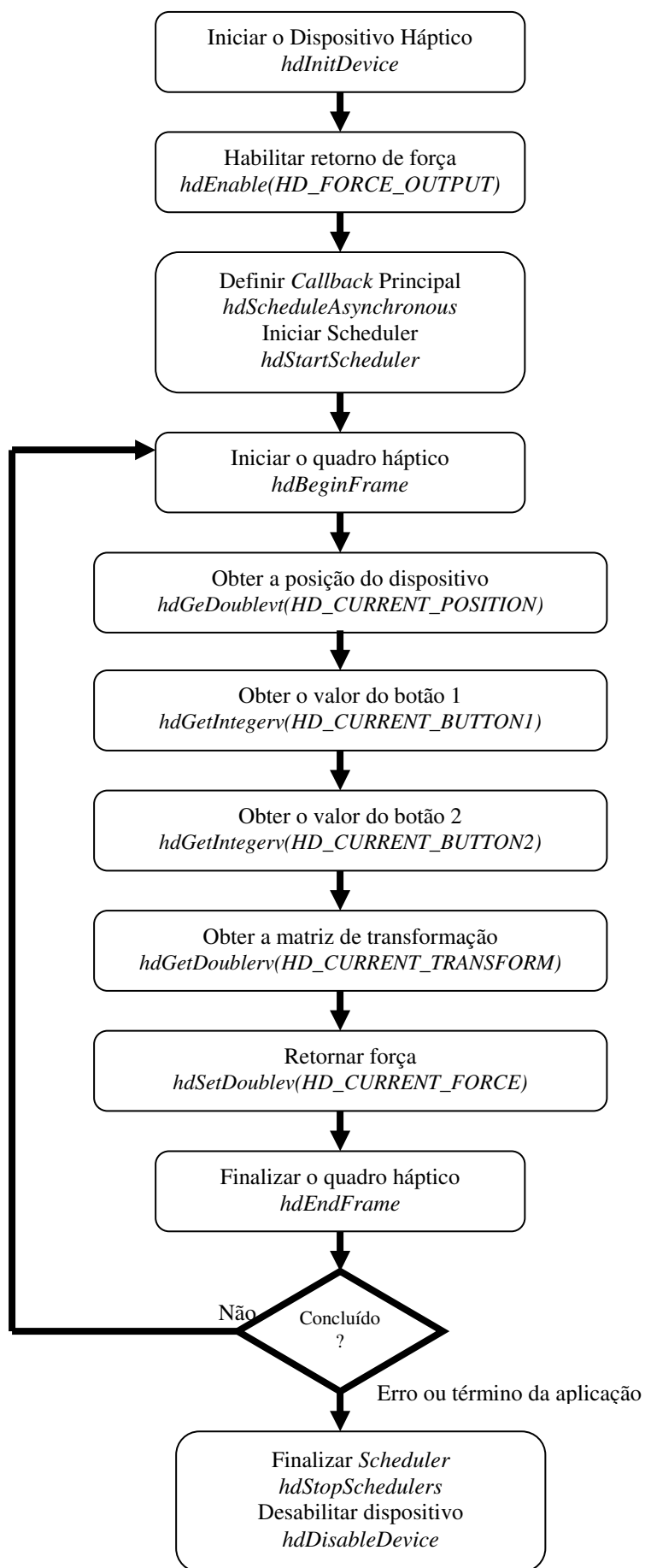


Figura 45 – Diagrama de execução para o dispositivo háptico

Para que se possa trabalhar com o dispositivo é necessário primeiramente inicializá-lo, o que é realizado com a função HDAPI *hdInitDevice*. Em seguida, o retorno de força é habilitado por meio da função *hdEnable (HD\_FORCE\_OUTPUT)*, e o *Scheduler* é definido como assíncrono (*hdSchedulerAsynchronous*). O *Scheduler* é responsável por acionar as *callbacks*, podendo ser síncronos ou assíncronos, com prioridades de execução definidas pelo usuário. Uma função *callback* consiste em uma *thread*, ou seja, um processo leve para manter a frequência de comunicação com o dispositivo em 1000 Hz. Uma *callback* definida como principal foi implementada e sua execução foi iniciada pelo comando *hdStartScheduler..*

Estas funções do *OpenHaptics Toolkit* são acionadas na execução do método em Java *initializeHapticDevice*, que retorna os valores para a variável *verify*, indicando consistências de dados do dispositivo. O valor 1 indica que o dispositivo está pronto para ser utilizado, -2 indica erro na inicialização do *Scheduler*, e -1 indica erro na inicialização do dispositivo.

A variável *verifyThread*, definida em um código nativo separado, e acionada pelo método em Java *verifyCallback*, verifica a execução da *callback* principal, sendo que o valor -1 indica erro na execução e o valor 1 indica que ela está sendo executada corretamente.

As funções foram inseridas dentro da *callback* principal, e são acionadas pelos métodos em Java *button1State*, *receiveTranslation*, *receiveData*, *sendData*, respectivamente. As funções do *OpenHaptics Toolkit* utilizadas nesse projeto são descritas na Tabela 14, juntamente com a descrição de cada comando, omitindo-se as variáveis de armazenamento, que acompanham os comandos, sendo definidas como parâmetros nas funções. Observa-se que os botões citados estão localizados no braço do dispositivo háptico, sendo o botão 1 de cor mais escura e o botão 2 de cor mais clara, conforme a Figura 46.



Figura 46 – Botões do PHANTOM *Omni*Tabela 14: Funções do *OpenHaptics Toolkit*

Descrição do Comando	Função
hdGetIntegerv(HD_CURRENT_BUTTON_1)	Obter o estado do botão 1
hdGetIntegerv(HD_CURRENT_BUTTON_2)	Obter o estado do botão 2
hdGetDoublev(HD_CURRENT_POSITION)	Obter a posição do dispositivo nos três eixos
hdGetDoublev(HD_CURRENT_TRANSFORM)	Obter uma matriz de transformação para realização de rotações
hdSetDoublev(HD_CURRENT_FORCE)	Realizar o retorno de força

As informações recebidas ou enviadas para essas funções HDAPI são armazenadas em variáveis e vetores globais, sendo os vetores definidos como ponteiros, que recebem e enviam informações para a parte da aplicação construída em Java por meio dos métodos. A Figura 81, encontrada no Apêndice C, apresenta o trecho de código da *callback* principal, escrita em linguagem de programação C++.

Para que a rotação do objeto que representa o instrumento médico ocorra de forma correta, foi usado um vetor de 16 posições para armazenar uma matriz de transformação contendo 4 linhas e 4 colunas (*transform[16]* em C++, e *datamatrix[16]* em Java). Essas informações são captadas por funções do dispositivo háptico, conforme mencionado, e enviadas para a aplicação desenvolvida em Java, onde as posições são alteradas, realizando a transposição da matriz.

A transposição consiste na inversão das linhas e das colunas de uma matriz quadrada, neste caso, realizando a inversão dos valores obtidos com a função do *OpenHaptics Toolkit* e armazenados em um vetor de 16 posições, conforme descrito no parágrafo anterior, propiciando a correta rotação e translação de um objeto virtual na execução do método *set* da classe *Transform3D*, implementado em Java3D. Este método possui como parâmetro, dentre outros, a matriz de transformação de 4 linhas e 4 colunas armazenados em um vetor de 16 posições.

No entanto, diversos vetores de três posições foram usados em Java e C++ para receber informações sobre translação (*translation[3]* em C++ e *translationdata[3]* em Java), velocidade nos eixos x, y e z (*velocity[3]* em C++ e *velocity[3]* em Java), e enviar informações, como o retorno de força nos três eixos (*force[3]* em C++ e *dataarray[3]* em Java). Além disso, variáveis do tipo inteiro foram usadas para determinar o pressionamento e a liberação dos botões do dispositivo. O esquema disponibilizado na Figura 47 apresenta a execução da aplicação, no que diz respeito à integração, relacionando a parte implementada em Java (quadros azuis) e a parte implementada em C++ (quadros vermelhos) descritas até o momento.

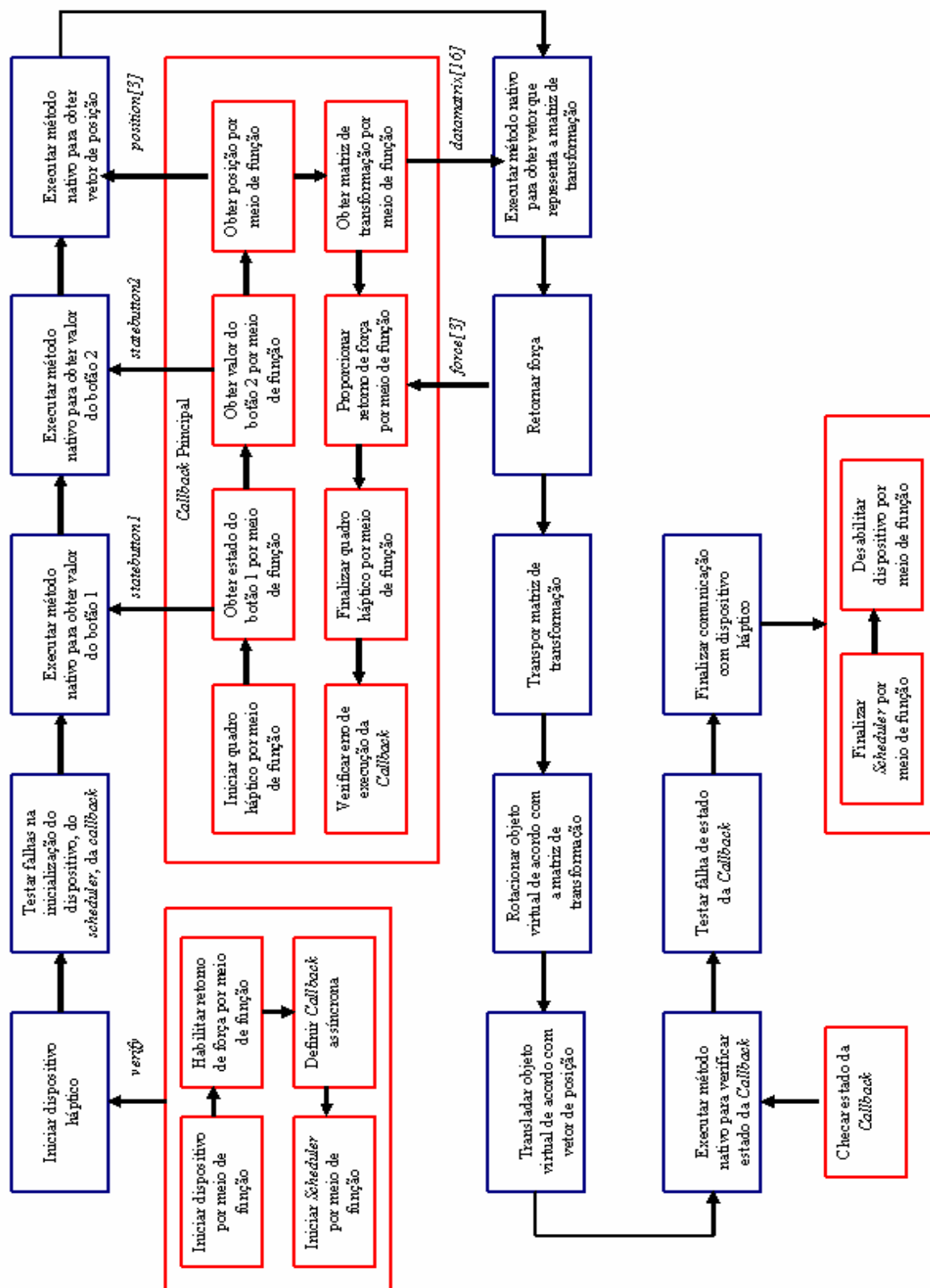


Figura 47 – Esquema de integração do dispositivo háptico

Todas as etapas descritas nesta seção são apresentadas no esquema anterior, incluindo-se, a seqüência de execução; as variáveis utilizadas, os testes para verificação de erros na inicialização do dispositivo háptico, do *Scheduler*, e durante a execução da função *Callback*; o momento em que os métodos nativos são executados acionando as funções nativas em C++, que executam funções HDAPI, indicando também quando o objeto virtual que representa o instrumento médico sofre as transformações de rotação e translação.

#### 4.9 Integração da Luva de Dados

O diagrama da Figura 48 apresenta as etapas do código em C++ para o dispositivo luva de dados. Primeiramente, ocorre a verificação das portas USB por meio do comando *fdScanUSB*, para verificar o local de conexão da luva de dados. Depois, é definido o tipo de luva ou luvas de dados conectadas, identificando o número de sensores, e o lado ao qual a luva de dados se refere (mão direita ou mão esquerda).

Com a função *fdOpen*, realiza-se o início da comunicação com o dispositivo, armazenando-se o endereço que representa a luva de dados em um ponteiro denominado *pGlove*, definido como sendo do tipo *fdGlove*. Se um erro ocorrer na inicialização do equipamento, a variável *verify* recebe o valor -1; caso contrário, esta variável recebe o valor 1, indicando que operações podem ser realizadas. As operações com a luva são realizadas utilizando este ponteiro como parâmetro nas funções fornecidas pelo fabricante.

A função *fdGetNumSensors* também é empregada para a obtenção do número de sensores do dispositivo, uma vez que o *software* pode ser utilizado por outras luvas de dados do mesmo fabricante. As funções em C++ foram inseridas em um bloco de código nativo, acionado pelo método *openGlove*, que retorna para a aplicação os valores 1 ou -1, indicando a presença ou ausência de erro.

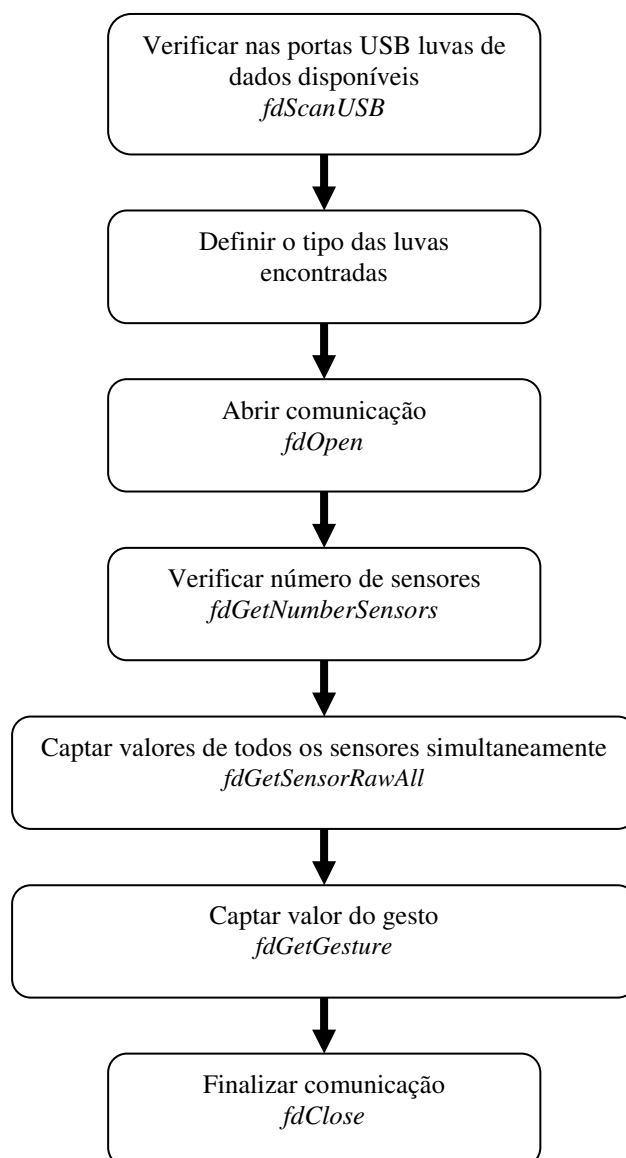


Figura 48 – Diagrama de execução para o dispositivo luva de dados

O trecho de código da Figura 83, encontrada no Apêndice D, mostra as implementações em C++ com toda a etapa de inicialização da luva de dados.

No código em linguagem de programação C++ outras funções foram implementadas para serem acionadas por métodos nativos. As funções *fdGetSensorRawAll* e *fdGetGesture* são adotadas para captar os valores de todos os sensores e o número de um gesto, respectivamente, sendo acionadas por dois métodos diferentes em Java, que foram colocados no bloco que realiza as operações a cada quadro. Essas funções são explicadas detalhadamente na Figura 84 do Apêndice D.

Os valores obtidos por meio das funções em linguagem de programação C++, são passados para um vetor declarado na classe *Glove*, escrita em Java, por meio de um método nativo. Cada valor do vetor é utilizado como parâmetro para definir o ângulo de inclinação a ser aplicado na esfera de rotação (base de cada dedo) e para rotacionar o conjunto de objetos que representa cada dedo. Para isso, é usado o método *setRotation* da classe *Transform3D*, classe que por sua vez é acionada pelo método *set* da classe *TransformGroup*, que está associado ao objeto virtual em questão (no caso, a esfera de rotação). Os detalhes da modelagem da mão virtual e, conseqüentemente, dos dedos, serão apresentados na seção 4.10.

A função *fdClose*, empregada para fechar a comunicação com a luva de dados, recebe como parâmetro o ponteiro *pGlove*, e é acionada na execução do método *finalize*, executado quando o objeto é destruído, e antes que o *Garbage Collector* seja acionado.

O esquema apresentado na Figura 49 mostra a execução da aplicação, no que diz respeito à integração, relacionando a parte implementada em Java (quadros azuis) e a parte implementada em C++ (quadros vermelhos) descritas nesta seção, indicando as etapas e a seqüência de execução; as variáveis utilizadas; o momento em que os métodos nativos são executados acionando as funções nativas em C++, que executam funções disponibilizadas pelo fabricante, indicando também quando os objetos virtuais sofrem transformações.

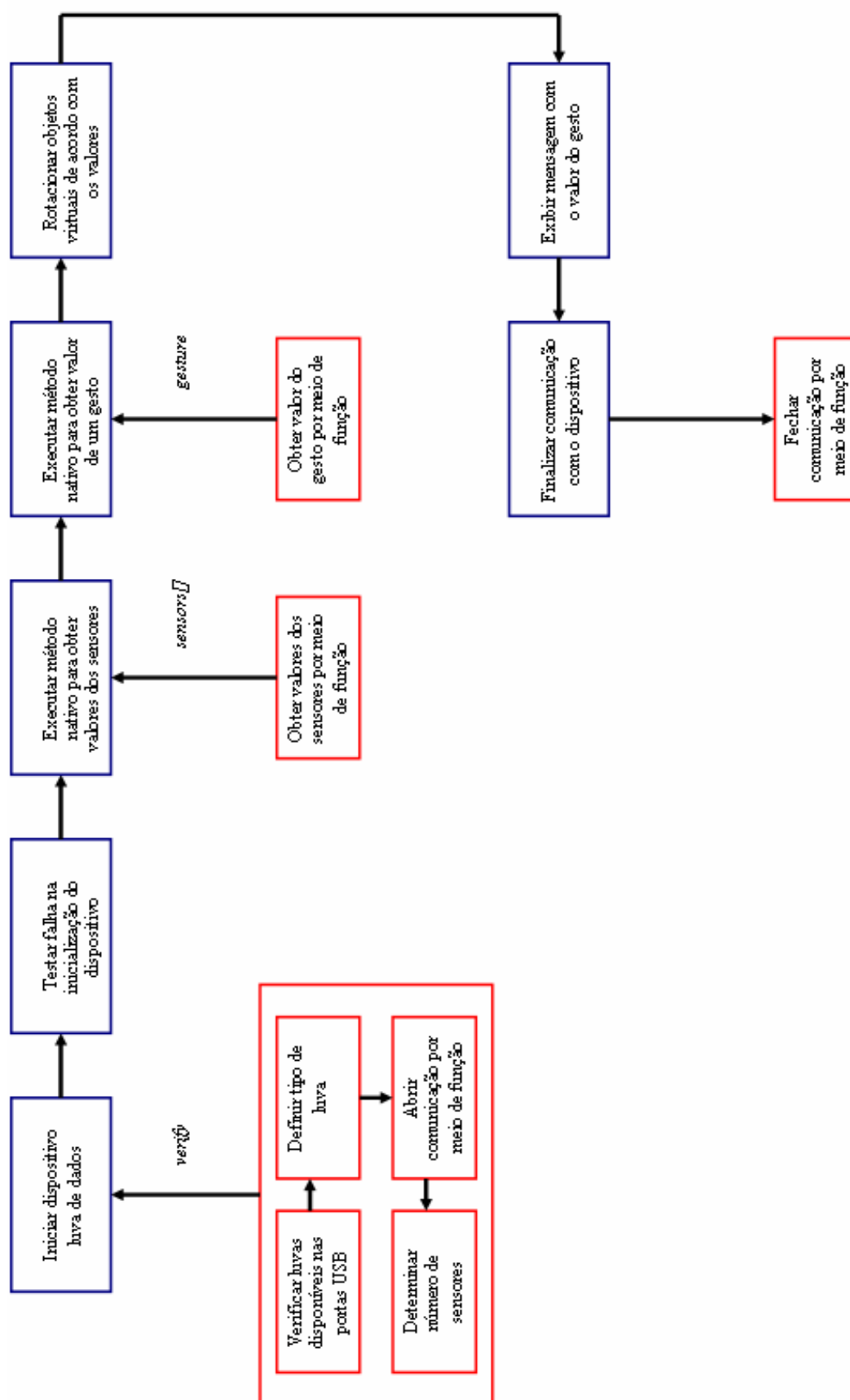


Figura 49 – Esquema de integração da luva de dados

## 4.10 Construção dos Objetos Tridimensionais

Outro aspecto importante da implementação do módulo de interação no ViMeT é a construção dos objetos tridimensionais que compõem o AV. Com relação à luva de dados, a modelagem de uma mão virtual foi projetada com uma estrutura, na qual os objetos que representam os dedos foram divididos, visando à obtenção de uma maior sensação de realismo, pois há uma hierarquia no movimento das falanges dos mesmos.

A Figura 50 mostra a divisão exata, cujos dedos são presos na palma da mão da seguinte forma: as pontas dos dedos são penduradas no corpo dos respectivos dedos, os corpos dos dedos são pendurados nas esferas de rotação, e as esferas são penduradas na palma da mão. Em testes realizados em outras aplicações, verificou-se que os dedos não rotacionavam corretamente, por isso a necessidade da colocação de esferas, permitindo que a rotação se limite ao centro de cada esfera, para propiciar um maior realismo.

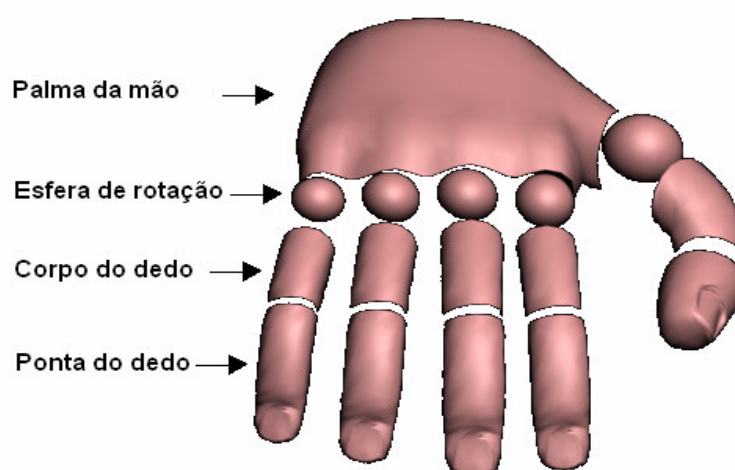


Figura 50 - Estrutura de modelagem da mão

Visando ainda a elevar o grau de realismo e devido ao fato da luva de dados utilizada no projeto possuir somente um sensor por dedo, optou-se por fazer as modelagens dos objetos que representam os dedos no AV de forma que eles apresentem um leve grau de flexão das



falanges mais distantes da esfera, ao contrário da forma reta usada inicialmente. A Figura 51 exemplifica as duas formas de modelagem usando um único dedo.

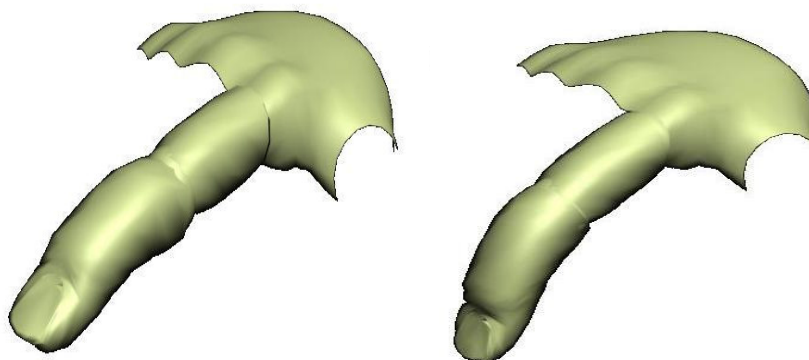


Figura 51 – Inclinação dos dedos virtuais

A Figura 85a (Apêndice E) apresenta o trecho de código em Java para importação e definição de características dos objetos virtuais que compõem a mão e a Figura 85b apresenta o código para a hierarquização dos objetos virtuais.

A seringa virtual também foi inserida em uma estrutura hierárquica, controlada por uma esfera colocada na ponta da mesma, de acordo com a Figura 52, permitindo que a detecção de colisão ocorra entre o centro da esfera e o objeto mama virtual. Dessa forma, é possível atingir uma precisão mais elevada, tornando os movimentos mais realistas quando executados pelo usuário por meio do dispositivo háptico. O código para definição e composição da seringa virtual é apresentado na Figura 86 (Apêndice E).

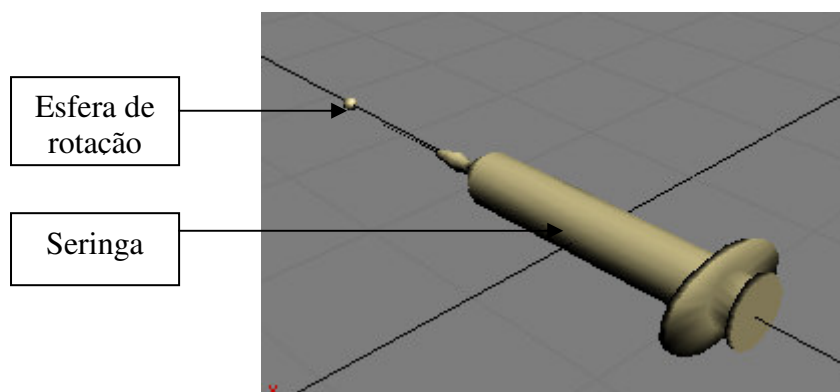


Figura 52 – Hierarquia de objetos na construção da seringa virtual

Devido a essas modelagens, foi necessário realizar alterações no grafo de cena para composição do AV para inserir as esferas e a mão virtual. A esfera permite que a rotação do objeto que representa o instrumento médico, no caso, a seringa, ocorra de forma correta, tomando como ponto de orientação, o centro da mesma. Isto também aumenta a precisão da colisão, visto que o centro do objeto é usado para determinar e refinar a aproximação e a colisão propriamente dita dos objetos virtuais.

Os grafos de cena inicial e atualizado podem ser visualizados nas Figuras 53, 54a e 54b. O grafo é iniciado com o *MyUniverse*, que representa o ambiente a ser criado. Em seguida o nó *MyLocale*, que indica onde serão posicionados os componentes do AV, cujos *BranchGraphs* (BGs) dividem os componentes do AV de acordo com suas características (objeto deformável (BG1), objeto rígido (BG2), por exemplo). Estes últimos são ligados aos *TransformGroups* (TGs), presentes em dois níveis de hierarquias para definir diferentes características, como a rotação, translação e escala em um nível, e estereoscopia em outro nível, por exemplo.

O nó BG3 representa a adição do dispositivo *mouse* no grafo de cena, adicionando-o no *MyLocale* e permitindo que diversos dispositivos possam ser adicionados. Os nós *Shapes* (*ShapeDef* e *ShapeRig*) representam os objetos virtuais a serem importados, e os círculos *Appearance* e *Geometry*, definem a aparência e a geometria dos objetos, respectivamente. Os nós *ViewPlatform*, *View*, *PhysicalBody*, *PhysicalEnvironment*, *Canvas3D*, *Screen3D* permitem a visualização do AV por parte do usuário.

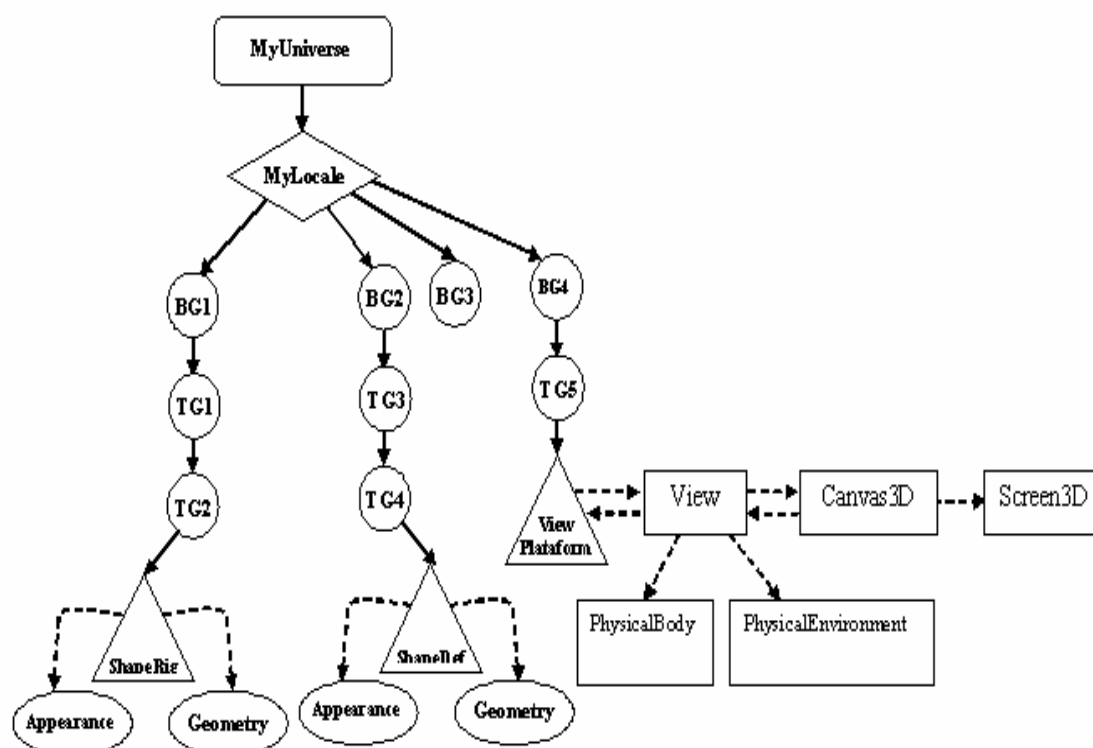


Figura 53 – Grafo de cena inicial

Para a construção da mão virtual, foram elaborados subgrafos, que foram sendo agrupados, constituindo a hierarquia dos dedos. As Figuras 53 e 54a apresentam o grafo de cena atualizado após a implementação do módulo de interação, no qual o nó BG3 pode representar a inclusão do dispositivo háptico ou luva de dados, lembrando que a adição de um outro nó do tipo BG permite a utilização de mais de um dispositivo simultaneamente. No lado esquerdo da Figura 53, pode-se observar a adição de um subgrafo (em cor vermelha), responsável pela definição do objeto virtual que representa o instrumento médico (BG1), o qual é conectado à outra estrutura, iniciada em TG1, responsável pela definição do objeto esfera, que por sua vez, é conectada ao *MyLocale*. No grafo anterior, o nó que representava o objeto que indicava o instrumento médico (BG1) era conectado diretamente ao *MyLocale*, visto que não havia a esfera de rotação. Esta mudança não é adotada quando a interação

ocorre por meio do *mouse*, somente quando o dispositivo em questão é o equipamento háptico.

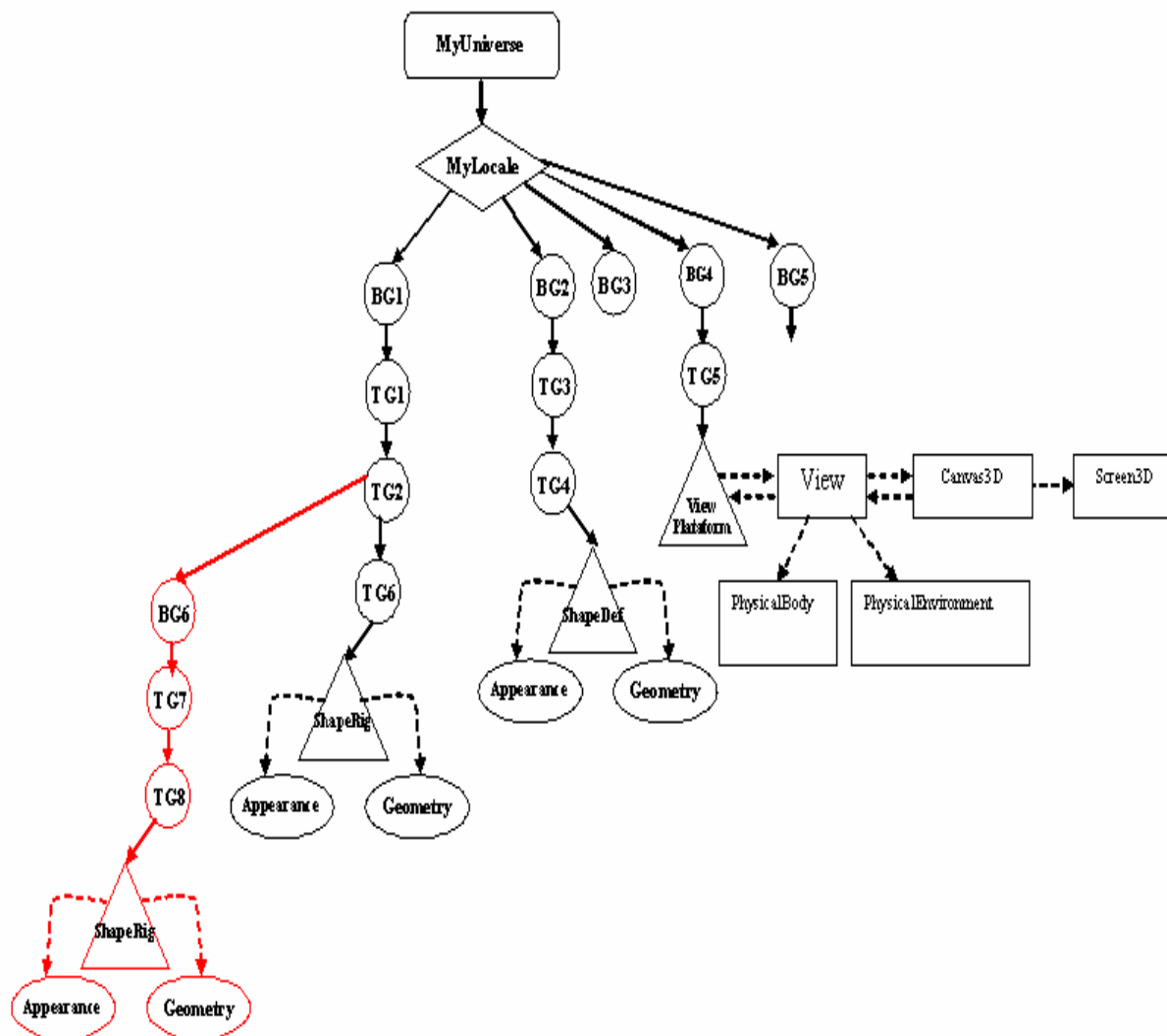


Figura 54a – Grafo de cena atualizado do ViMeT

A continuação do BG5 é mostrada na Figura 54b, definindo a hierarquia de objetos no grafo de cena para confecção do dedo, a palma da mão é representada pela estrutura superior central, e os BGs conectados a esta estrutura representam os subgrafos para a composição hierárquica de cada objeto que formará cada dedo, visto que os dedos podem ser considerados um conjunto de objetos ou um conjunto de subgrafos.

Pode-se observar nesta figura, que para cada dedo existem três objetos virtuais conectados entre si, formando subgrafos de forma seqüencial, no qual, partindo-se de cima para baixo, logo após o subgrafo central, o primeiro representa a esfera de rotação (iniciado em BG10), o segundo o objeto modelado que representa o corpo do dedo (iniciado em BG12), e o terceiro o objeto modelado que representa a ponta do dedo (iniciado em BG13).

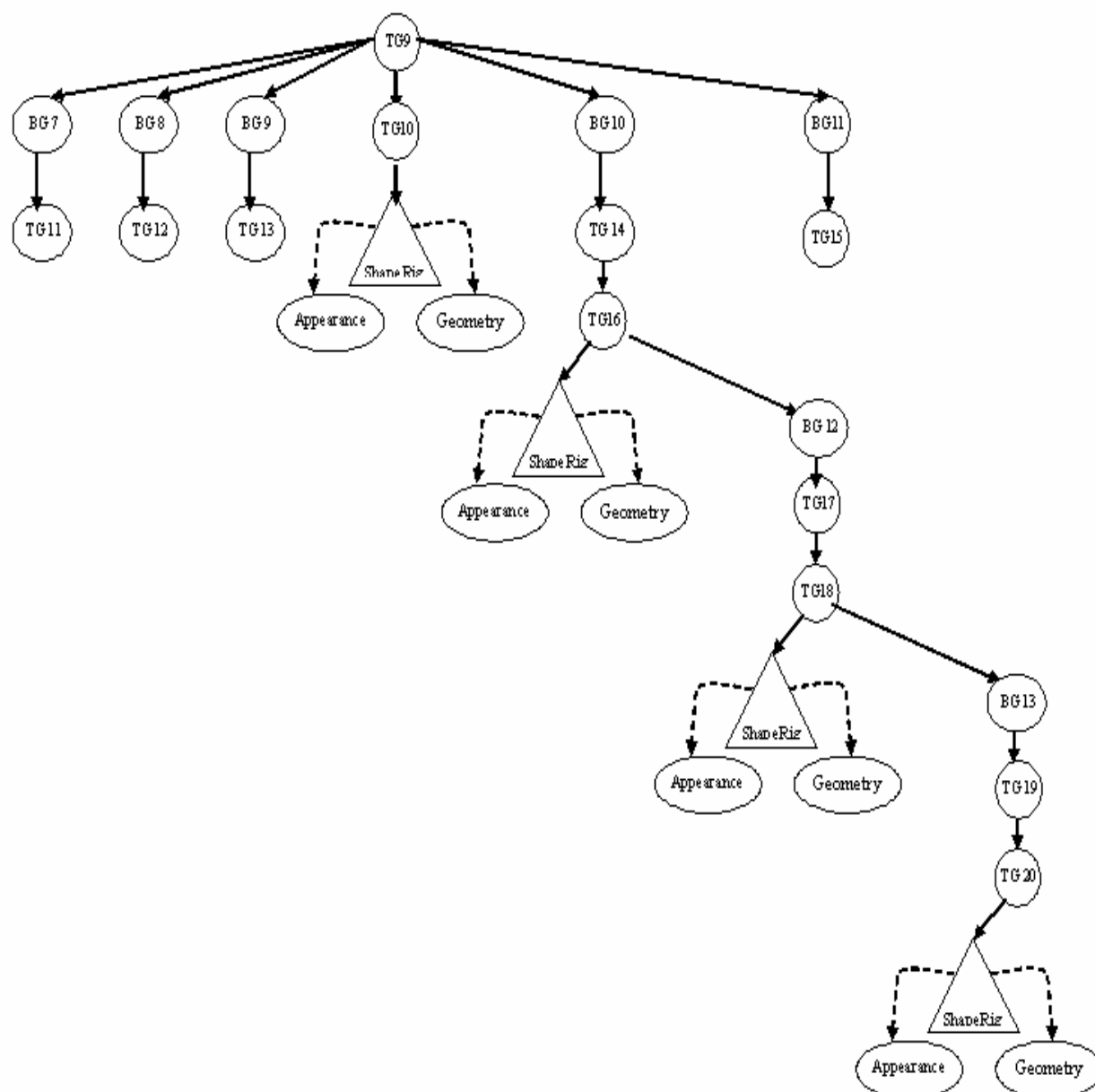


Figura 54b – Continuação do grafo de cena atualizado

Outro detalhe importante é que a aplicação deve ser responsável pela importação dos objetos virtuais, que são definidos pelo usuário durante a geração da mesma, o qual também precisa definir algumas características, como: posição, rotação e tamanho dos objetos, bem como os dispositivos a serem utilizados. As posições da palma da mão, dos dedos e da esfera são definidas na geração da aplicação, sem a intervenção do usuário nesta etapa. O usuário pode alterar tais parâmetros na aplicação gerada. Um exemplo do novo AV construído pelo ViMeT após a implementação do módulo de interação é mostrado na Figura 55.

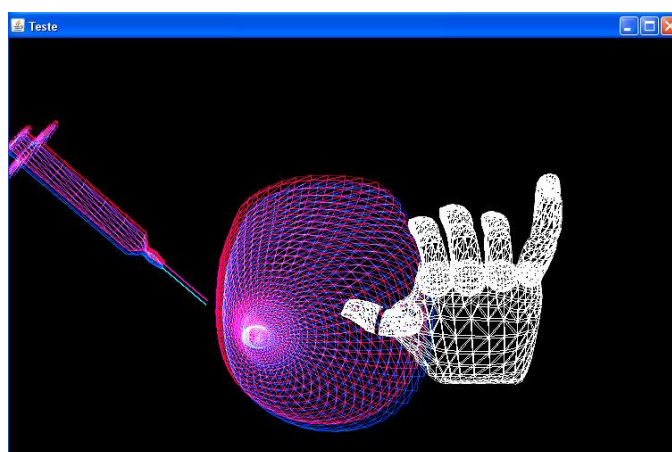


Figura 55 – Novo AV gerado por meio do ViMeT

#### 4.11 Integração com o ViMeT

O módulo de interação com suporte a dispositivos convencionais e não convencionais foi incorporado ao ViMeT. A utilização do módulo de interação ao ViMeT ocorre pela instanciação das classes relacionadas a cada dispositivo na aplicação gerada (*Haptic*, *Glove*, *Mouse* e *Keyboard*), com os *TransformGroups* e objetos *Object3D* representando os objetos virtuais a serem associados aos seus respectivos dispositivos e sendo passados como parâmetros, como o trecho de código apresentado na Figura 56.

```

//Dispositivo Háptico
Haptic h = new Haptic(objetos[2].getMotionTransform(),
cd, super.myLocale);

//Luva de dados
Glove l = new Glove(objetos[17],objetos[16],
                    objetos[15],objetos[14],objetos[13],
                    super.myLocale);

//Mouse
Mouse m = new Mouse(objetos[2].getMotionTransform(),
super.myLocale);

//Teclado
Keyboard k = new Keyboard(objetos[17],objetos[16],
                          objetos[15],objetos[14],objetos[13],
                          super.myLocale);

```

Figura 56 – Trecho de código para incorporação do módulo de interação

No caso da luva de dados e do teclado, 5 objetos *Object3D* são passados como parâmetro, representando as esferas de rotação para movimentação dos dedos da mão virtual. No caso do dispositivo háptico, o *TransformGroup* associado à esfera (associada ao objeto que representa o instrumento médico) é passado como parâmetro para receber as transformações de acordo com os dados do equipamento. Com relação ao *mouse*, o *TransformGroup* é associado diretamente ao objeto que representa o instrumento médico, e é passado como parâmetro para sofrer transformações durante a interação.

Na passagem de parâmetros com o dispositivo háptico pode-se notar a presença do objeto *cd*, instanciado da classe *Collision*, para indicar em que momentos durante a interação são necessários o cálculo e o retorno de força (na presença ou ausência de colisão).

Todas as classes possuem o parâmetro *MyLocale*, onde o subgrafo (*BranchGraph*) criado no método construtor de cada uma das classes é adicionado por meio do método *addBranchGraph*, sendo que tais classes contêm as características e funcionalidades para realizar operações com os referidos dispositivos. Desta forma, diversos dispositivos podem ser adicionados simultaneamente na aplicação.

A ferramenta de instanciação automática *ViMeTWizard* também foi modificada para permitir a escolha da combinação de dispositivos a ser utilizada na aplicação a ser gerada, conforme mostra a Figura 57, na qual o usuário pode optar por *mouse* ou dispositivo háptico e teclado ou luva de dados, disponibilizados na guia *Devices*. Pode-se observar também os objetos definidos pelo usuário para a simulação e a ausência da mão virtual, visto que essa última é importada automaticamente, com características pré-definidas (posição, orientação e escala). Por esse motivo, algumas vezes tornam-se necessárias modificações no código gerado.

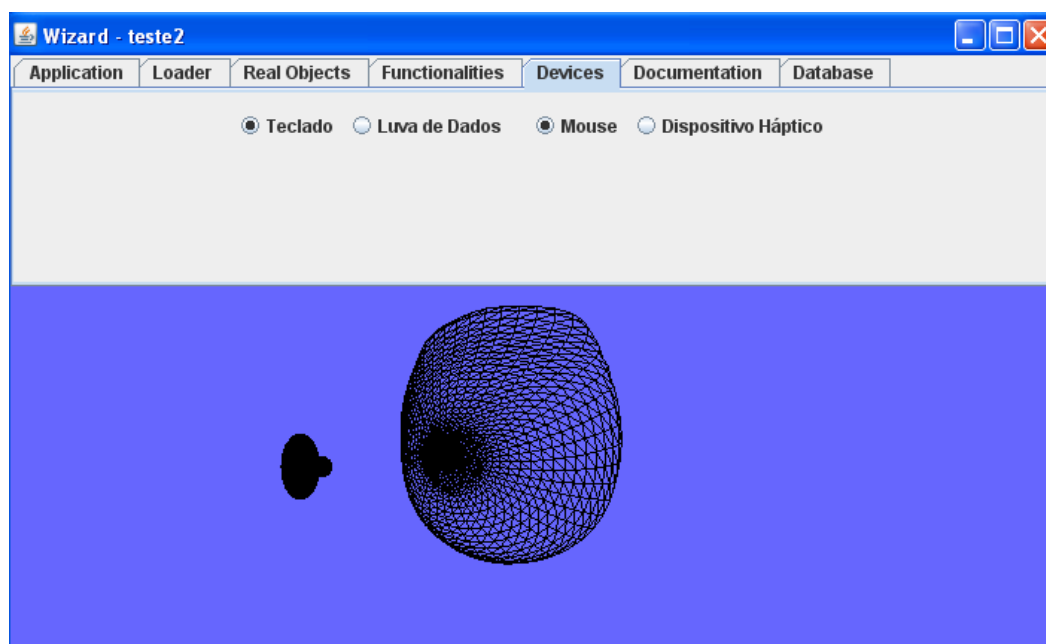


Figura 57 – Tela para seleção de dispositivos na *ViMeTWizard*

#### 4.12 Avaliação de Interação em Aplicações Geradas a Partir do ViMeT

A segunda etapa deste trabalho foi a avaliação da interação em aplicações construídas com o ViMeT, levando em consideração aspectos computacionais, como o tipo de computador e o tipo de aplicação, procurando verificar o desempenho do sistema, como o



tempo de resposta diante de um comando do usuário e o número de quadros gerados por segundo.

Além disso, os dispositivos utilizados são determinantes, pois envolvem questões como número de graus de liberdade, eficiência no processamento, maior ou menor grau de realismo, sendo que estas questões estão interligadas. Uma quantidade maior de graus de liberdade de um dispositivo, por exemplo, pode significar um número maior de cálculos ou instruções para execução, influenciando na eficiência do processamento. A camada adicional para a integração entre as linguagens de programação Java e C++ realizada neste trabalho, deve ser levada em consideração no que se refere ao processamento.

Os aspectos relacionados aos usuários também foram analisados, como o conforto no uso dos dispositivos, a satisfação do usuário, a combinação desejável de dispositivos (*mouse* e teclado ou luva e equipamento háptico), o tempo de execução de uma determinada tarefa, as experiências anteriores com aplicações de RV e com o procedimento médico a ser executado (exames de punção).

#### **4.12.1 Metodologia de Avaliação**

Primeiramente, foi realizada uma descrição oral do processo de avaliação para os voluntários, definindo suas etapas, objetivos, e explicando também conceitos de RV, como definições sobre dispositivos convencionais e não convencionais, equipamento háptico, bem como algumas peculiaridades do sistema e dos dispositivos.

Uma avaliação preliminar com um professor da área médica definiu que a execução de parte do procedimento de biópsia seria realizada separadamente para cada dispositivo. O motivo para este procedimento foi a observação de que a mão virtual fixa no AV dificultava a

visualização do usuário, já que a luva de dados não dispõe de um sensor de posição para captar movimentos do usuário.

O tempo de execução de uma determinada tarefa foi analisado, no caso, parte do exame de biópsia. Com o *mouse* ou o equipamento háptico, o usuário deveria manipular o instrumento médico virtual tentando tocar o objeto virtual que representa o órgão humano.

Para completar a avaliação foi analisado o desempenho da aplicação, sendo mensurado o número de *frames* por segundo e o tempo de resposta. O tempo de resposta é o tempo decorrido entre um comando do usuário e a ação do sistema, exibindo um movimento do objeto virtual que representa o instrumento médico no monitor de vídeo.

Dois questionários para a coleta dos dados foram elaborados, um para ser respondido antes e outro após a avaliação (Apêndice A). O primeiro continha questões que tratavam do conhecimento relativo às áreas de RV e Medicina, no que diz respeito à execução de exames de biópsia, bem como a opinião sobre sistemas de RV em saúde. O segundo foi constituído por questões sobre a composição do AV, a aparência dos objetos virtuais, o conforto e a facilidade de uso com cada dispositivo.

Para cálculos de tempo de resposta e número de quadros por segundo, foram utilizadas classes estendidas da classe *Behavior*, com métodos *initialize* e *processStimulus*, e a classe *WakeupOnElapsedFrames*, que propicia a criação de um *loop* de simulação, conforme o trecho de código da Figura 58. Como *WakeupOnElapsedFrames* com parâmetro igual a zero, define que o código escrito no método *processStimulus* seja executado a cada *frame*, foram inseridos dentro deste método variáveis e comandos para obter o tempo de resposta e o número de quadros por segundo. Os métodos para acionar as funções em código nativo são colocados neste bloco, permitindo que as informações entre dispositivos e aplicação sejam

intercambiadas em tempo real. Desta forma, a cada ação do usuário, o sistema realiza uma reação, propiciando a interação.

```

0001
0002     public void initialize()
0003     {
0004         wAct = new WakeupOnElapsedFrames(0); //Execução a cada frame
0005         wakeupOn(wAct);
0006     }
0007
0008     public void processStimulus(Enumeration criteria)
0009     {
0010         ...

```

Figura 58 – Trecho de código da classe *Behavior*

O tempo de resposta é calculado pela divisão entre 1000 milissegundos e o número de *frames* gerados por segundo pela aplicação. Para cálculo do número de quadros por segundo são usadas três variáveis, *timeend*, *timebegin* e *framenumber*. A cada execução do bloco de código do método *processStimulus*, a variável *framenumber* é incrementada. Quando o valor da variável *timeend* é igual ou maior a 1000, equivalente ou próximo a um segundo, um novo tempo inicial é recebido pela variável *timebegin* e a variável *framenumber* é exibida, apresentando o número de quadros, conforme apresentado na Figura 59. Este trecho de código foi inserido na classe *Behavior* associada à classe *Haptic*, para propiciar a interação com o dispositivo háptico a cada *frame*, sendo que o trecho de código da linha 4 até a linha 11 é inserido no início do método *ProcessStimulus*, e o trecho das linhas 14 e 15 é inserido no final do referido método.

```

0001
0002
0003     //Verifica se o tempo decorrido foi de 1 segundo
0004     if (timeend >= 1000.0)
0005     {
0006         //Exibe a quantidade de frames por segundo
0007         System.out.println("Numero de frames - Haptico: " + framenumber);
0008
0009         timebegin = System.currentTimeMillis();
0010         framenumber = 0;
0011     }
0012
0013     //Incrementa a variável a cada iteração
0014     framenumber++;
0015     timeend = System.currentTimeMillis() - timebegin;

```

Figura 59 – Trecho de código para coletar informações

### **4.13 Considerações Finais**

Este capítulo apresentou a construção das classes em Java para cada um dos dispositivos, a integração das linguagens de programação Java e C++, e os detalhes da implementação, ferramentas e recursos utilizados, além de problemas encontrados e soluções adotadas, trechos do código nas linguagens de programação C++ e Java.

No próximo capítulo, serão apresentados também os métodos, comandos e variáveis para mensurar o tempo de resposta e a quantidade de quadros por segundo, usados na avaliação, bem como uma exposição sucinta dos questionários, que podem ser encontrados no Apêndice A.

## CAPÍTULO 5 – RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados alcançados no que diz respeito à implementação da interação no ViMeT, e as informações coletadas na avaliação da interação.

A implementação foi realizada para os quatro dispositivos citados, construindo o módulo de interação com suporte aos dispositivos convencionais (*mouse* e teclado), e não convencionais (luva de dados e dispositivo háptico), fazendo para os dois últimos, a integração das linguagens de programação C++ e Java.

A avaliação com usuários da área de aplicação foi realizada objetivando obter parâmetros para aperfeiçoar a aplicação gerada pelo ViMeT e auxiliar na construção de novas aplicações dessa mesma natureza. Com as implementações realizadas foi possível disponibilizar uma forma flexível de interação para as aplicações geradas pelo ViMeT, com as seguintes características:

- **navegação:** realizada pelas setas do teclado, possibilitando que o usuário, segundo uma das classificações da categoria de navegação, realize a exploração do ambiente, observando e analisando os objetos virtuais (órgão humano e instrumento médico). Neste contexto, a navegação também pode ser classificada como *travel*, uma vez que o usuário poderá movimentar-se pelo mundo virtual, alterando seu ponto de vista;

- **seleção e manipulação:** não há ocorrência de seleção durante a sessão de interação, visto que os objetos virtuais a serem manipulados estão associados aos dispositivos, como o instrumento médico, que está associado ao *mouse* e ao dispositivo háptico; e a mão virtual, a qual está associada à luva de dados e ao teclado. Pode-se dizer que a seleção ocorre na geração e definição da aplicação, momento em que os objetos virtuais são associados aos

dispositivos para serem manipulados. Em relação à manipulação, é permitida a alteração de tamanho, posição, orientação e rotação dos objetos citados. Com todos os dispositivos (convencionais e não convencionais), é possível movimentar o objeto virtual que representa o instrumento médico, o que é realizado com o *mouse* e o dispositivo háptico, cada qual com suas características e limitações, e os dedos do objeto virtual que representa a mão do usuário, o que é realizado com a luva de dados (flexionando os dedos), e o teclado (pressionando e liberando teclas específicas);

- **controle do sistema:** o controle é realizado automaticamente, uma vez que o usuário pode proceder a manipulação dos objetos virtuais por meio dos dispositivos, e realizar a navegação por meio das setas do teclado, sem utilizar por exemplo, teclas específicas para indicar uma mudança de controle.

Neste módulo, no que diz respeito às formas de interação (direta, controles físicos e controles virtuais), é utilizada a forma direta quando trata-se da luva de dados modificando a orientação dos dedos virtuais e a forma realizada através de controles físicos, pois dispositivos são empregados para navegar pelo ambiente, manipular objetos virtuais e controlar o sistema (MINE, 1995).

As seções seguintes descrevem o funcionamento do módulo de interação e a avaliação realizada junto à Famema (Faculdade de Medicina de Marília).

## 5.1 Resultados da Implementação

Como descrito anteriormente, a implementação da interação envolveu a construção de um módulo de interação com dispositivos convencionais e não convencionais para executar determinadas operações no AV.

Com o uso do teclado, é possível navegar pelo mundo virtual, bastando recorrer às teclas que representam as setas, podendo, desta forma, verificar a composição do AV, os objetos virtuais e seus comportamentos. Além disso, este dispositivo é usado para alterar a rotação dos dedos da mão virtual, representados por teclas específicas que, quando pressionadas ou liberadas, determinam a movimentação dos mesmos, e representam o ato de segurar e soltar o órgão humano durante o procedimento médico. As teclas escolhidas foram L, K, J, H e B, para uso da mão direita do usuário, em um formato que indique que a mão está em posição de segurar e soltar um determinado objeto. No entanto, outras teclas podem ser escolhidas e definidas na classe *Keyboard*. A Figura 60 apresenta a interação por meio do teclado, movimentando os dedos da mão virtual.

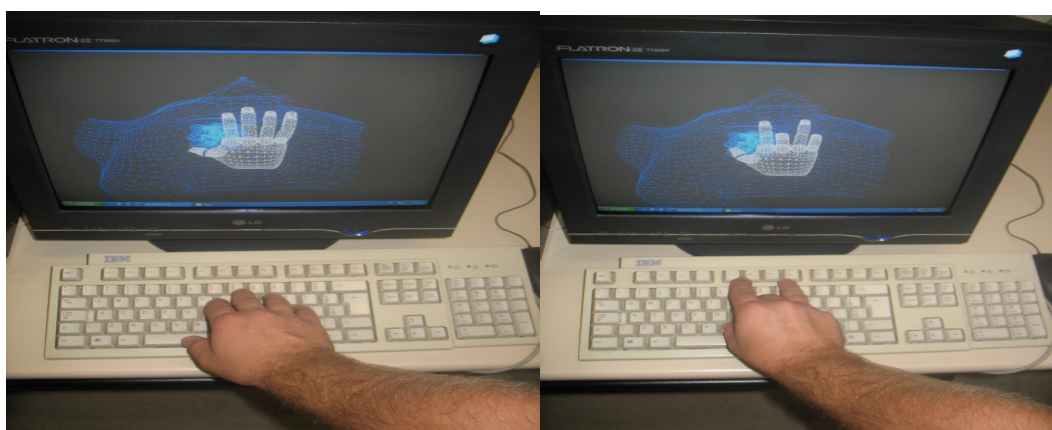


Figura 60 – Exemplo de utilização do dispositivo teclado

Tanto o *mouse* quanto o dispositivo háptico movimentam o instrumento médico virtual de acordo com os movimentos do usuário. Entretanto, o dispositivo háptico trabalha com um número maior de informações (translação e rotação nos três eixos), do que o *mouse* (movimentos de translação em dois eixos). Além disso, o dispositivo háptico retorna força ao usuário, causando sensações táteis a este quando a aplicação detecta uma colisão, indicando que o instrumento usado no procedimento tocou o órgão humano virtual. Neste momento do treinamento, uma deformação do objeto órgão humano é realizada pela aplicação e

apresentada na tela do monitor. A Figura 61 apresenta a interação por meio do *mouse* comum, que permite ao usuário manipular o objeto virtual que representa o instrumento médico (no caso, a seringa virtual).

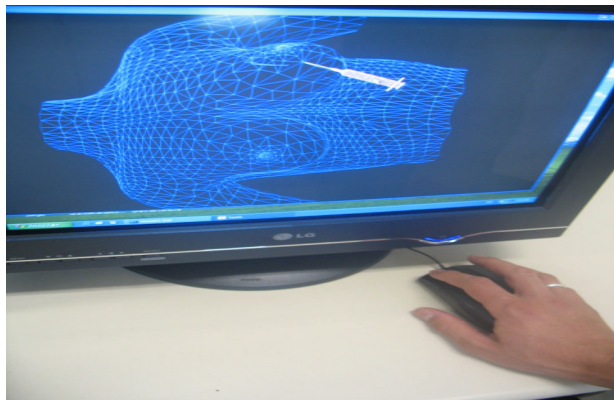


Figura 61 – Exemplo de utilização do dispositivo *mouse*

A Figura 62 apresenta a utilização do dispositivo háptico na manipulação do instrumento médico.

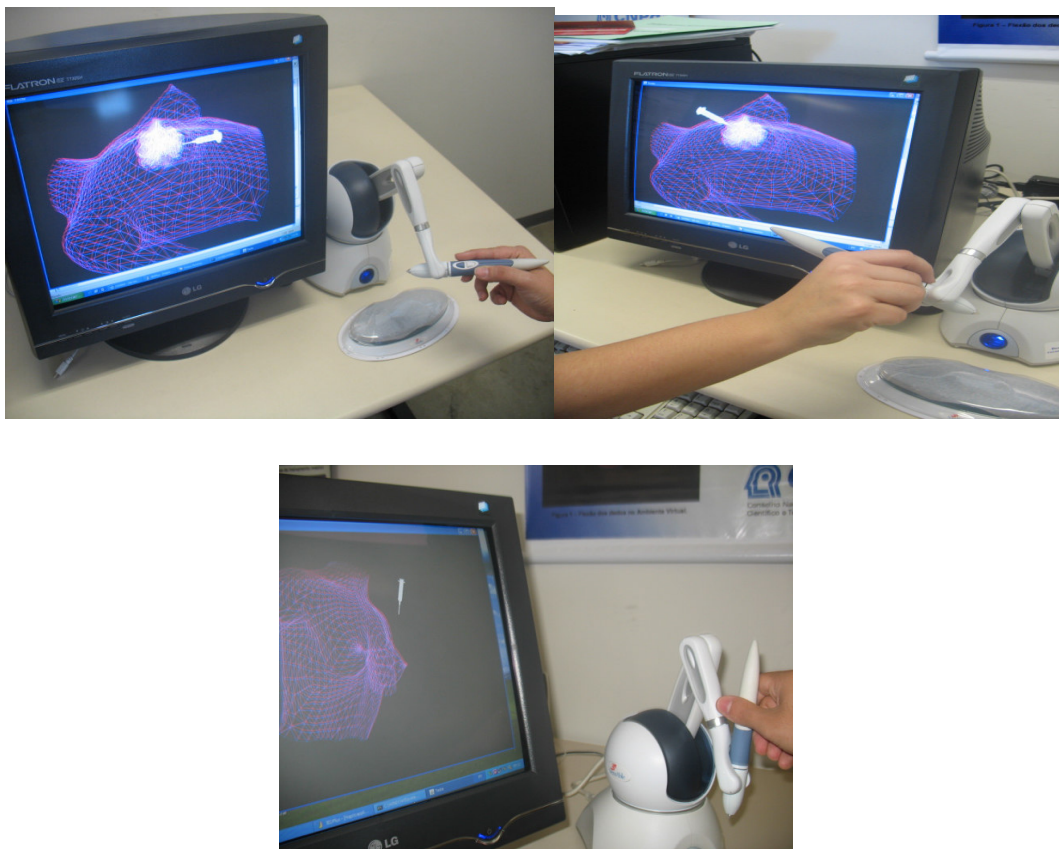


Figura 62 – Exemplos de utilização do dispositivo háptico



A luva de dados, por sua vez, representa o ato de segurar e soltar o órgão por meio da mão virtual, da mesma forma que o teclado. Os dedos da mão virtual são rotacionados de acordo com a flexão captada pelos sensores da luva, conforme a Figura 63, que apresenta as modificações causadas pela aplicação na mão virtual quando os dedos do usuário são flexionados. A translação da mão virtual não pôde ser implementada, visto que a luva de dados não possui um detector de posição da mão do usuário; entretanto, um detector de posição poderia resolver este problema.

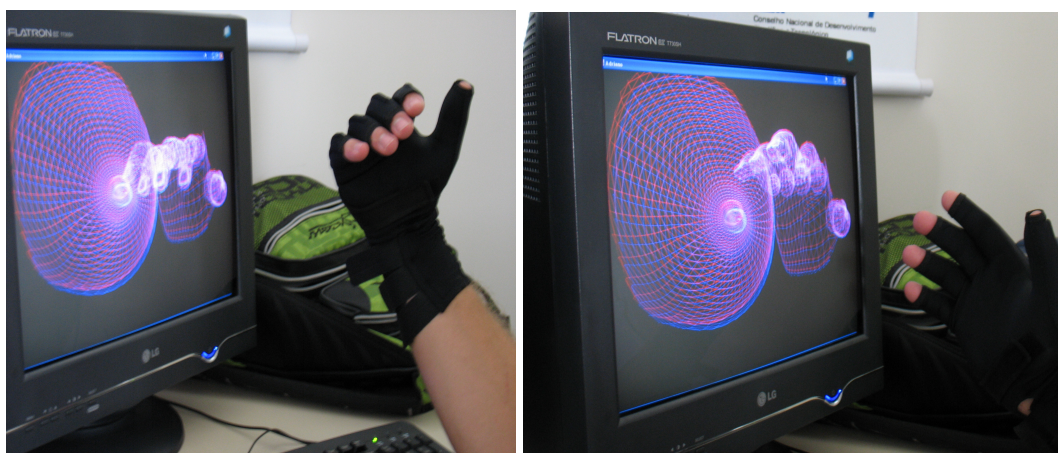


Figura 63 - Demonstração de movimentos captados pela luva de dados

Nos sistemas de RV, principalmente naqueles relacionados ao treinamento médico, é desejável, entre diversas características, precisão e respostas em tempo real, para que a interação durante o treinamento seja mais próxima da realidade (NUNES *et al.*, 2007a), oferecendo experiências úteis aos profissionais de Medicina. Um AV para simular procedimentos médicos, de acordo com MACHADO *et al.* (2004e), apresenta vantagens, como:

- disponibilidade do AV, propiciando a minimização da necessidade de obter cobaias ou pacientes reais toda vez que o treinamento precisar ser realizado;

- inexistência de riscos a pacientes, médicos e estudantes da área, pois pessoas não dotadas da habilidade necessária podem cometer erros, causando danos a determinados tecidos e ferimentos aos pacientes, por exemplo;

- diversidade de casos clínicos, pois casos normais ou com anomalias podem ser apresentados, bem como situações encontradas em exames de biópsia no cotidiano de profissionais que executam tais procedimentos.

Entretanto, atualmente o treinamento não está totalmente implementado nas aplicações geradas pelo ViMeT. Ele é concluído quando o usuário toca o órgão humano com o instrumento médico por meio dos dispositivos, gerando a deformação. A finalização do exame, consistindo na retirada do instrumento médico, e o depósito do material coletado em local apropriado ainda deverão ser implementados no ViMeT.

Em relação às vantagens citadas anteriormente, verifica-se que as aplicações geradas pelo ViMeT propiciam as seguintes vantagens: disponibilidade do AV, visto que a sessão de treinamento pode ser realizada toda vez que o usuário desejar executar a aplicação; e eliminação de riscos a pacientes, médicos e estudantes de Medicina, que podem realizar, até o presente momento, parte do procedimento de biópsia. No entanto, ainda não é fornecida uma diversidade de casos e não estão implementadas algumas questões importantes, como as diferentes camadas da pele causando diferentes sensações táteis.

## **5.2 Avaliação de Profissionais da Área Médica**

A avaliação por profissionais da área médica ocorreu em duas etapas: a primeira no dia 26 de junho de 2008 e a segunda no dia 3 de julho de 2008, nas dependências da Famema. A primeira foi realizada com um professor da área médica, e a segunda com alunos do curso de

Medicina e um monitor do laboratório de informática da instituição. Na segunda etapa foram aplicados dois questionários: um que deveria ser preenchido antes do treinamento (denominado questionário pré-teste) e outro após o treinamento (denominado questionário pós-teste).

Na primeira etapa, o professor testou aplicações com diferentes combinações de dispositivos (*mouse* e teclado, *mouse* e luva de dados, dispositivo háptico e teclado, dispositivo háptico e luva de dados), bem como cada dispositivo isoladamente. Realizou testes com visualizações em *wireframe* e com texturas coloridas, sendo uma aplicação com textura apresentada na Figura 64.

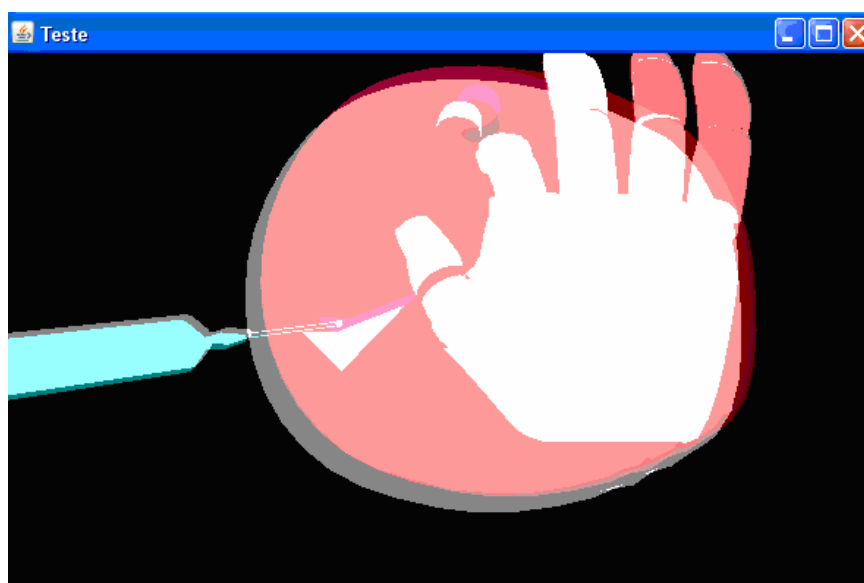


Figura 64 – Deformação dos objetos com cor

Com base nas opiniões deste profissional, algumas alterações foram realizadas na implementação. A visualização com cor, por exemplo, não foi satisfatória, visto que dificultava a percepção espacial do AV e prejudicava o realismo, principalmente na ocorrência da deformação. Por isso, a visualização com objetos virtuais em aramado foi adotada. A Figura 65 apresenta a deformação gerada pela aplicação na visualização em aramado.

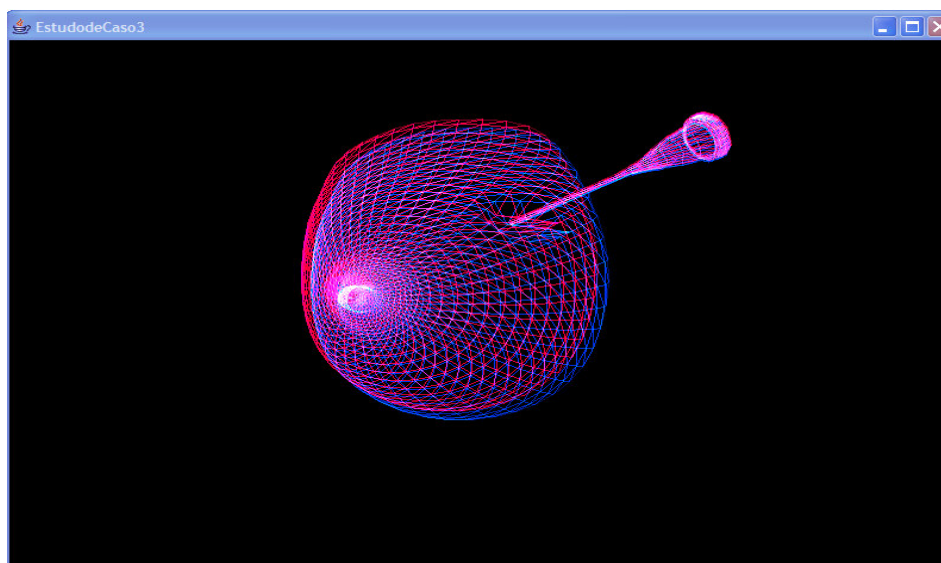


Figura 65 – Deformação em aramado

Uma tentativa de correção da correlação espacial também foi realizada, visto que os valores recebidos do dispositivo háptico para translação nos eixos x, y e z, são maiores para o AV criado em Java3D. Por isso, os valores fornecidos pelo dispositivo háptico foram divididos por uma constante definida empiricamente (20, no caso). Este valor empírico foi escolhido com base em testes com valores aleatórios, para determinar a posição do objeto que representa o instrumento médico no AV, compatível com a API Java3D, conforme o trecho de código da Figura 66. A nomenclatura *t3d* indica um objeto *Transform3D*, ligado ao *TransformGroup*, com o método *setTranslation*.

```
t3d.setTranslation(newVector3d
(translationdata[0]/20,translationdata[1]/20,translati
ondata[2]/20));
```

Figura 66 – Trecho de código da translação do dispositivo háptico

A construção de um modelo tridimensional mais elaborado, como um tórax, por exemplo, em complementação à mama, objeto tridimensional usado no primeiro teste, também foi sugerida, e um novo modelo foi construído. A Figura 67 apresenta o AV gerado

com o objeto virtual mama, antes da primeira avaliação, e a Figura 68 apresenta um tórax modelado com a presença de dutos internos, depois da referida avaliação.

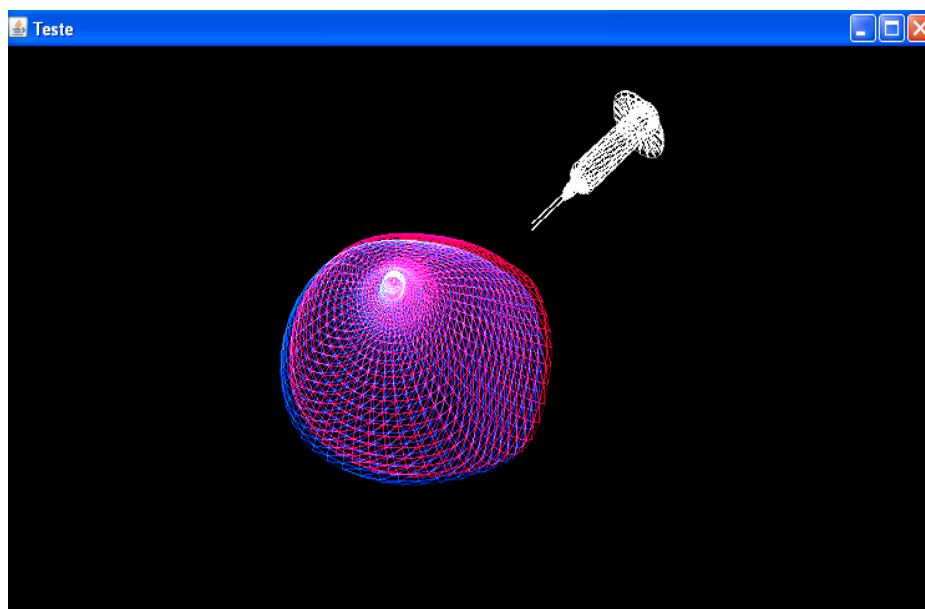


Figura 67 – Aplicação gerada anteriormente à construção do módulo

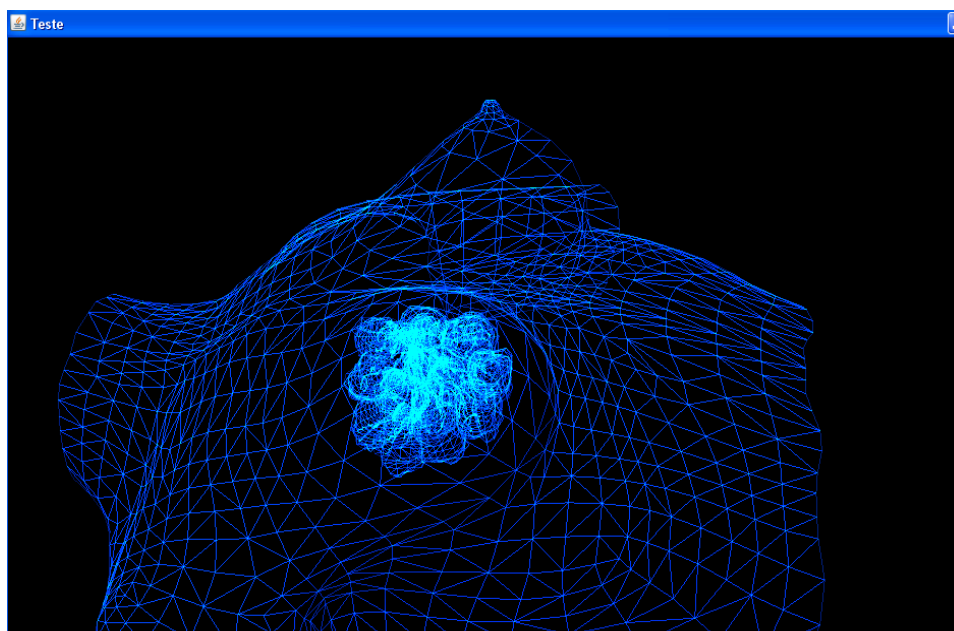


Figura 68 – Modelagem do tórax com os dutos internos

Outra questão foi com relação à mão virtual, que estando fixa no AV prejudicava a visualização, visto que a luva de dados não possui um sensor de posição e o sistema não dispõe de um rastreador de posição. Desta forma, decidiu-se que a segunda fase da avaliação

ocorreria com os dispositivos isoladamente, e no AV seriam importados somente os objetos envolvidos em cada momento do treinamento. Por exemplo, quando o *mouse* ou o equipamento háptico estivessem sendo usados, a mão virtual não seria carregada no ambiente sintético, somente o órgão humano e o instrumento médico, conforme é mostrado na Figura 69.

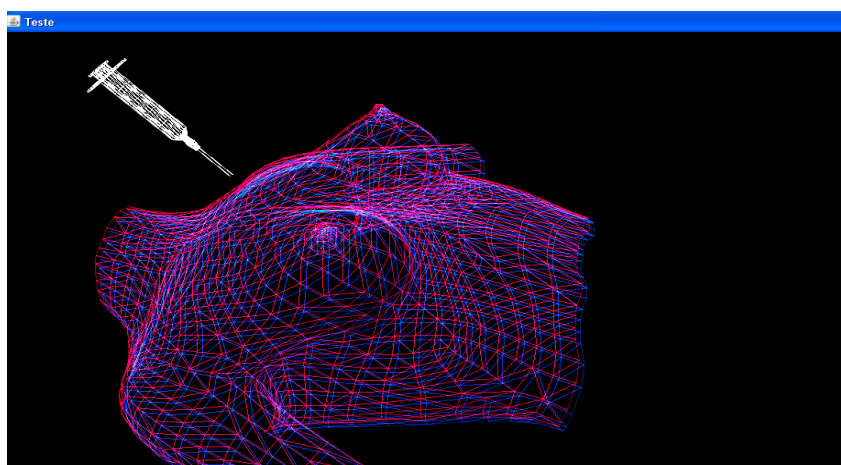


Figura 69 – Modelagem tridimensional de um tórax

A Figura 70 apresenta o AV com o tórax e a mão virtual, utilizados na simulação com o teclado e a luva de dados. Como é possível observar nas Figuras 68, 69 e 70, as novas modelagens incluem objetos virtuais mais completos para o órgão humano, contribuindo para o aumento do realismo.

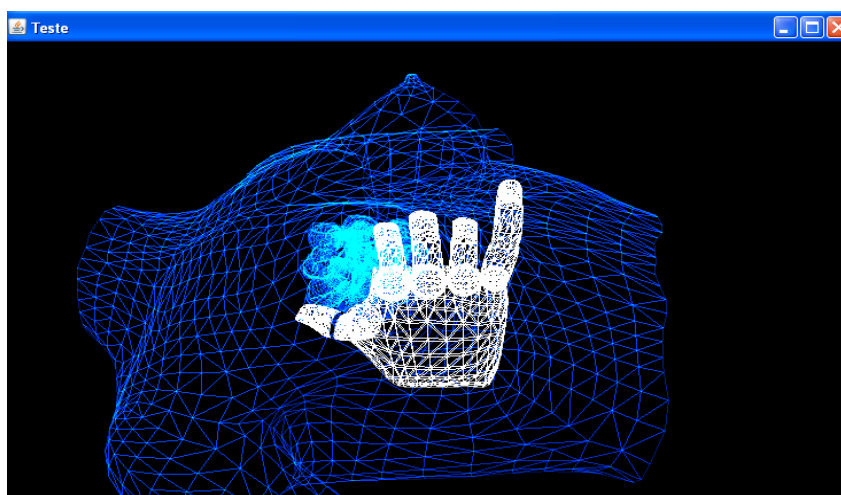


Figura 70 – Tórax e mão

Outras sugestões foram tecidas, que poderão ser consideradas em versões futuras do ViMeT: a correlação espacial mais precisa entre o objeto virtual (instrumento médico) e o dispositivo háptico; a inclusão de outros pontos de vista com o intuito de melhorar a percepção do ambiente; o acoplamento de uma seringa ou agulha real no braço do equipamento háptico para elevar o realismo; a inclusão de um objeto físico para representar o órgão humano no equipamento háptico ou próximo a este, com o intuito de facilitar a percepção visual.

Na segunda etapa, houve a presença de estudantes do curso de Medicina, sendo que 12 pessoas (11 estudantes e 1 monitor do laboratório de informática), com média de idade de 23 anos, preencheram o questionário pré-teste (Apêndice A).

Todos os entrevistados conheciam a área de RV, sabendo inclusive conceitos sobre algumas definições e aplicações, sendo que dois deles já tinham usado dispositivos de RV (óculos estereoscópicos). No entanto, nenhum voluntário havia, até aquele momento, trabalhado ou participado de simulações que envolviam luva de dados e equipamento háptico.

Desta forma, os onze participantes restantes estudavam Medicina, e três deles executavam procedimentos de biópsia, com experiências de 1 mês, sem especificação do número de procedimentos realizados, 2 meses, com média de 10 procedimentos executados, e 3 meses, com 12 procedimentos realizados.

Respondendo ao questionário pré-teste, todos disseram que simuladores de RV podem ser utilizados no aprendizado de procedimentos médicos, como exames de biópsia, sendo que dois ressaltaram a necessidade do aperfeiçoamento de tais aplicações. Essa questão foi repetida no questionário pós-teste.

Dos voluntários que participaram das simulações de treinamento, nem todos trabalharam com os quatro dispositivos. Pôde-se notar uma certa dificuldade para se adaptar

ao sistema e problemas para perceber a posição do objeto a ser manipulado no AV, principalmente no que diz respeito à profundidade.

A Tabela 15 apresenta os tempos de execução da tarefa para cada usuário, com os dispositivos *mouse* e equipamento háptico, mensurados em milissegundos. Para computar os tempos foram inseridos na classe *Octree* do ViMeT, responsável pela detecção de colisão, comandos para obter a hora do sistema durante a instanciação desta classe, quando a aplicação começa a ser carregada, e a hora do sistema no momento que a colisão entre objetos virtuais ocorre. Assim, o tempo inicial é subtraído do tempo final, resultando no tempo de execução da tarefa. Pôde-se observar que alguns usuários apresentaram mais de um tempo de execução, o que ocorreu devido à continuação do procedimento, após a colisão ter sido detectada, gerando outras colisões.

Tabela 15: Tempos de execução da tarefa em milissegundos

<b>Usuário</b>	<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>	<b>06</b>	<b>07</b>
<b>Dispositivo</b>							
<i>Mouse</i>	113.504	5.793	44.675 75.915				
<b>Equipamento Háptico</b>	15.302	10.507 12.901 25.303	22.437 28.231	30.462 31.022 33.461	18.325	40.525	20.600

Somente seis estudantes e o monitor do laboratório de informática preencheram o questionário pós-teste, uma vez que alguns voluntários tiveram que abandonar a sessão de avaliação em virtude de compromissos. Deste questionário, foi retirada a questão de número 7, que tratava da combinação de dispositivos, visto que sessões de testes com dois dispositivos simultaneamente não ocorreram. Entretanto, determinadas alternativas das questões 5 e 6, que também tratavam da combinação de dispositivos, deveriam ter sido alteradas, o que não aconteceu, gerando dificuldades em algumas respostas por parte dos voluntários. Um desses estudantes acompanhou os testes, mas não participou das simulações, não respondendo a determinadas questões.



Dos seis voluntários que participaram dos testes, 1 não se adaptou a nenhum dos dispositivos não convencionais (equipamento háptico e luva de dados), 5 apresentaram dificuldades com o dispositivo háptico, mas não tiveram problemas com a luva de dados. Isso pode ter ocorrido devido à complexidade da tarefa a ser realizada com o equipamento háptico, consistindo em manipular o objeto que representa a seringa até tocar no objeto que representa o órgão humano, quando comparada à tarefa a ser realizada com a luva de dados, que consistia na flexão dos dedos, para simular o ato de segurar e soltar o órgão.

Nenhum dos voluntários teve dúvidas quanto à tarefa a ser desempenhada no AV (parte de um procedimento de biópsia), ressaltando que esta foi explicada de forma clara, de acordo com a questão 2 do questionário pós-teste.

Sobre a facilidade de uso de cada dispositivo, em uma escala de 1 a 4, sendo que o dispositivo que apresentou maior facilidade deveria receber o valor 1, o segundo dispositivo o valor 2, e assim sucessivamente, dos 6 voluntários:

- 4 consideraram a luva de dados, 1 considerou o teclado, e 1 considerou o dispositivo háptico como nível 1 (mais fácil de usar);

- 3 consideraram o *mouse*, 2 consideraram o dispositivo háptico, 1 considerou o teclado como nível 2;

- 2 consideraram o dispositivo háptico, 1 considerou a luva de dados, 1 considerou o teclado, e 2 consideraram o *mouse* como nível 3;

- 2 consideraram o dispositivo háptico, 1 considerou a luva de dados, 1 considerou o teclado e 2 consideraram o *mouse* como nível 4.

Os gráficos da Figura 71 apresentam os dispositivos do módulo de interação e os níveis de facilidade de uso extremos na escala adotada (1 e 4), de acordo com as opiniões dos voluntários.

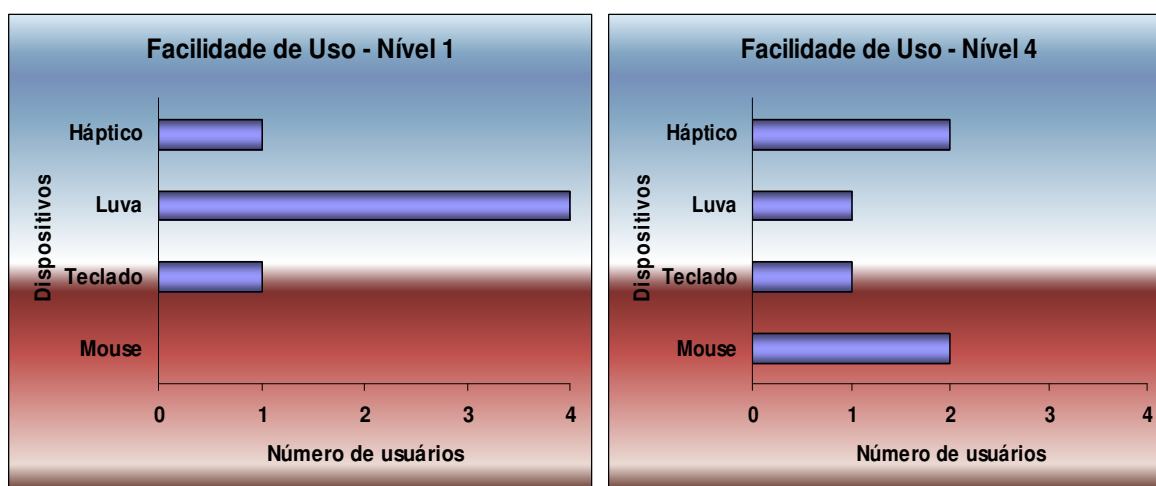


Figura 71 – Gráficos representando os dispositivos e o nível de facilidade de uso

A luva de dados foi considerada a mais fácil de usar pela maioria, possivelmente pelo tipo de tarefa executada com esse dispositivo. Entretanto, o gráfico da Figura 72 mostra que o dispositivo háptico apresentou uma facilidade de uso ligeiramente maior quando comparado com o *mouse*, sendo que os dois dispositivos são utilizados para manipulação do objeto virtual que representa o instrumento médico.

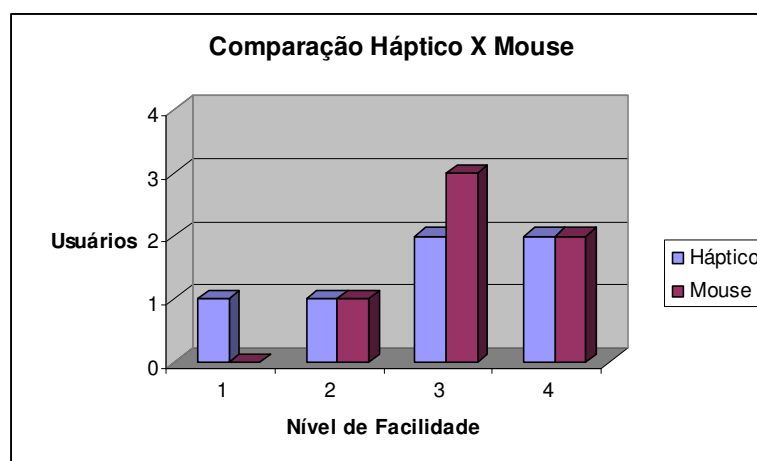


Figura 72 – Gráfico dos dispositivos associados ao instrumento médico

Com relação à luva de dados e ao teclado, o nível de facilidade da luva de dados foi bem maior, sendo ambos responsáveis pela flexão dos dedos da mão virtual, conforme o gráfico apresentado na Figura 73.

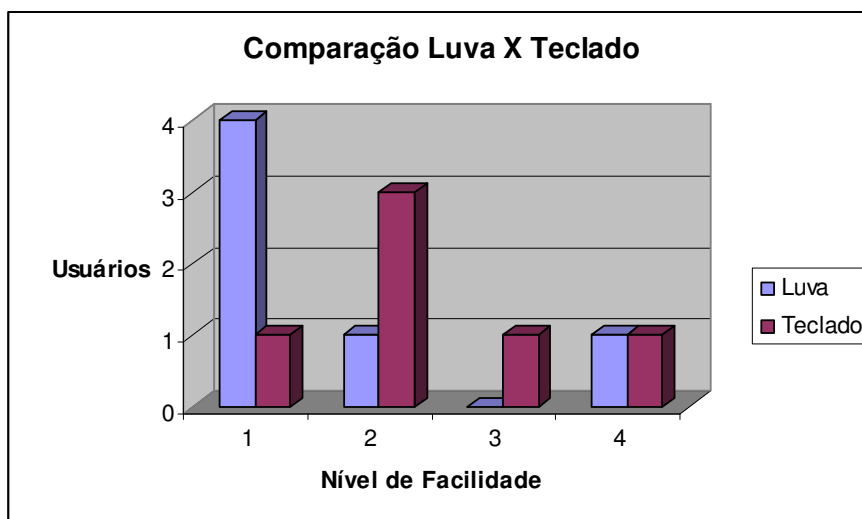


Figura 73 – Gráfico dos dispositivos associados à mão virtual

Sobre o aspecto visual do AV, levando em consideração a aparência e o comportamento dos objetos virtuais:

- 4 entenderam que o aspecto visual teve importante papel na simulação de treinamento;

- 3 entenderam que o aspecto visual do AV pouco contribuiu para a simulação.

Esta questão foi respondida pelo estudante que não participou dos treinamentos, mas acompanhou as sessões, visualizando inclusive o AV em um equipamento multimídia, disponibilizado pela Faculdade de Medicina. O gráfico da Figura 74 apresenta os resultados com relação à contribuição do aspecto visual na interação, com grande, pouca ou nenhuma contribuição. O objetivo é verificar o quanto a composição visual do AV influencia na realização do treinamento médico, além de avaliar a influência dos dispositivos.

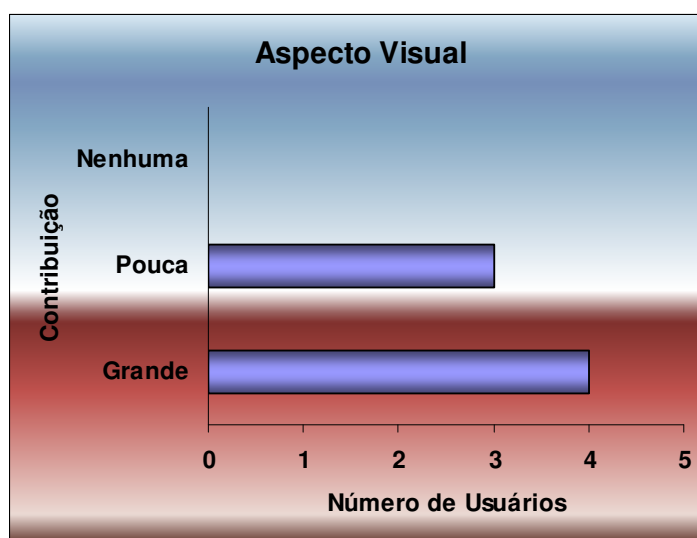


Figura 74 – Resultados do aspecto visual

No que diz respeito à utilização de dispositivos convencionais:

- 1 voluntário disse que eles tiveram importante papel na realização da tarefa durante a simulação;
- 2 disseram que eles pouco contribuíram para a realização da tarefa;
- 3 disseram que eles contribuíram satisfatoriamente quando foram combinados com os dispositivos não convencionais. Acredita-se que a primeira premissa seja verdadeira (os dispositivos convencionais contribuíram satisfatoriamente), entretanto, a combinação com dispositivos não convencionais, tratada na segunda premissa, não ocorreu.

O gráfico da Figura 75 mostra a opinião dos voluntários quanto à utilização dos dispositivos convencionais.

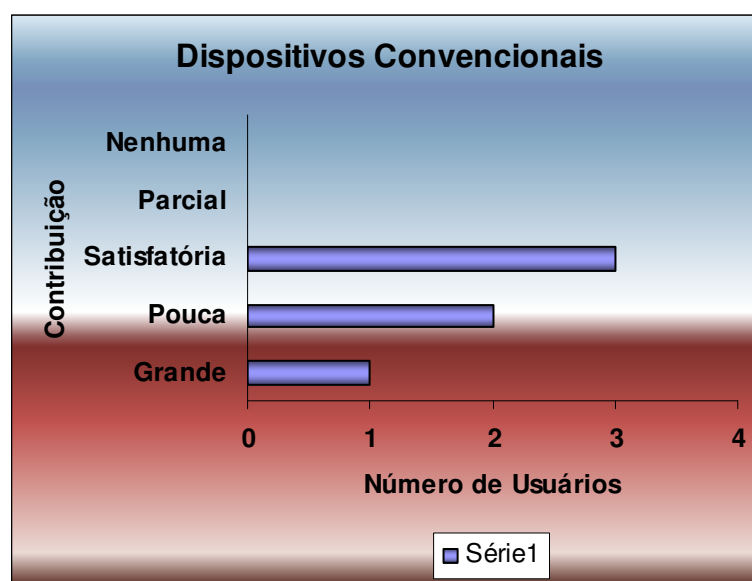


Figura 75 – Resultados com os dispositivos convencionais

Com relação à utilização de dispositivos não convencionais:

- 4 voluntários responderam que eles foram determinantes na realização do treinamento simulação;

- 2 disseram que tais equipamentos contribuíram satisfatoriamente para a realização da simulação de treinamento. Acredita-se que ocorreu fato semelhante à questão anterior, no que se refere à combinação com dispositivos, no caso, com dispositivos convencionais.

O gráfico da Figura 76 demonstra a opinião dos voluntários quanto à utilização dos dispositivos não convencionais.

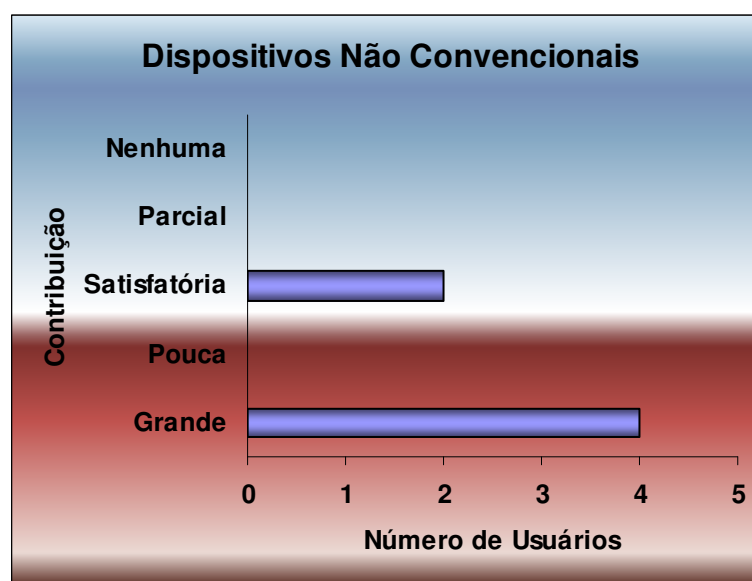


Figura 76 – Resultados com os dispositivos não convencionais

Na questão sobre a utilização de simuladores no auxílio ao aprendizado de procedimentos médicos, 2 disseram que eles podem ser considerados de grande importância neste sentido, e 5 disseram que eles podem auxiliar, mas precisam ser aperfeiçoados. Esta questão foi apresentada no questionário pré-teste, observando que 4 voluntários mudaram de idéia, o que ocorreu devido ao simulador ser um protótipo, estando em fase de implementação e aperfeiçoamento.

Segundo os voluntários, simuladores virtuais podem ser empregados:

- na aprendizagem de fisiologia e anatomia;
- desde simulações de pequenas cirurgias até anestésias, aplicações de vacinas e outras cirurgias mais complexas;
- outros procedimentos invasivos;
- no aprendizado de técnicas incisivas;
- na aplicação de injeções, punções, e na identificação da textura de tecidos patológicos.

Os voluntários também apresentaram idéias, sugestões e críticas, que podem ser utilizadas em trabalhos futuros, como o aperfeiçoamento da parte gráfica do AV, para tornar o treinamento menos difícil para o usuário; o desenvolvimento de formas para promover a sensibilidade da pele ao toque devido à existência de diferentes camadas; a inclusão de uma ou duas telas com outras perspectivas do AV, para melhorar a noção espacial; a correlação espacial entre o dispositivo físico e o objeto virtual associado, definindo uma escala de tamanho e movimentação entre a realidade e o AV.

As Tabelas 16 e 17 mostram o tempo de resposta e o número de quadros por segundo durante a utilização de cada um dos quatro dispositivos, usando dois computadores com configurações específicas. Os tempos de resposta e o número de quadros gerados por segundo atingiram valores satisfatórios, proporcionando a sensação de realismo nos testes realizados com os dois computadores. Segundo Kirner (2006), uma taxa de quadros por segundo ideal deve ser igual ou superior a 20 para manter a ilusão de movimento, sendo que uma taxa de 8 a 10 quadros por segundo é considerada aceitável para propiciar a imersão.

Tabela 16: Desempenho da aplicação para os dispositivos em um PC

<b>PC</b>	<b>Tempo de Resposta (ms)</b>	<b>Número de Quadros por Segundo</b>
<i>Mouse</i>	16,39	61
Teclado	16,66 a 16,94	59 a 60
Luva de Dados	16,39 a 17,54	57 a 61
Dispositivo Háptico	16,39 a 16,94	59 a 61

Tabela 17: Desempenho da aplicação para os dispositivos em um *Notebook*

<b><i>Notebook</i></b>	<b>Tempo de Resposta (ms)</b>	<b>Número de Quadros por Segundo</b>
<i>Mouse</i>	47,61	21
Teclado	50 a 58,82	17 a 20
Luva de Dados	52,63 a 71,42	14 a 19
Dispositivo Háptico	16,66	60

Conforme mostrado nas Tabelas 16 e 17, o sistema atingiu uma quantidade de quadros por segundo suficiente para proporcionar a sensação de resposta em tempo real. Pode-se observar que em um computador mais potente, a taxa de quadros por segundo é maior, no entanto, com a implementação futura de métodos de colisão e deformação mais elaborados e a utilização de objetos virtuais mais complexos para tornar a aplicação mais próxima do real, a taxa deve ser menor, mas é previsto que continue em nível satisfatório. Desta forma, o aumento no processamento reflete na taxa de quadros por segundo, mas dependendo do computador, não prejudica a interação, que a princípio deve continuar em nível adequado.

Um teste semelhante ao realizado com cada dispositivo individualmente foi feito para cada uma das combinações de equipamentos disponibilizados até o momento com o módulo de interação. As Tabelas 18 e 19 apresentam o tempo de resposta e o número de quadros por segundo para cada combinação de dispositivos executada. Novamente, os tempos de respostas e os números de quadros por segundo atingiram valores suficientes para imprimir sensação de realismo.

Tabela 18: Desempenho da aplicação para as combinações em um PC

<b>PC</b>	<b>Tempo de Resposta (ms)</b>	<b>Número de Quadros por Segundo</b>
<i>Mouse</i> e Teclado	16,39 a 18,86	53 a 61
<i>Mouse</i> e Luva de Dados	17,54 a 21,73	46 a 57
Dispositivo Háptico e Teclado	16,39 a 16,94	59 a 61
Dispositivo Háptico e Luva de Dados	16,66 a 19,23	52 a 60

Tabela 19: Desempenho da aplicação para as combinações em um *Notebook*

<b><i>Notebook</i></b>	<b>Tempo de Resposta (ms)</b>	<b>Número de Quadros por Segundo</b>
<i>Mouse</i> e Teclado	50 a 62,50	16 a 20
<i>Mouse</i> e Luva de Dados	52,63 a 58,82	17 a 19
Dispositivo Háptico e Teclado	55,55 a 62,50	16 a 18
Dispositivo Háptico e Luva de Dados	58,82 a 76,92	13 a 17



### 5.3 Considerações Finais

Os resultados apresentados atingiram um nível satisfatório em relação ao aspecto computacional, com tempos de resposta e número de quadros por segundo que proporcionaram uma interação em tempo real com cada dispositivo do módulo, bem como com cada combinação de dispositivos. Esses tempos podem ser utilizados para refinar os métodos de deformação, colisão e retorno de força, permitindo uma precisão mais elevada, cujo requisito é fundamental em aplicações relacionadas ao treinamento médico.

No aspecto humano, as informações coletadas nos questionários foram importantes para realizar alterações na aplicação, aperfeiçoando o *framework* e definindo os caminhos a serem seguidos em implementações futuras.

## CAPÍTULO 6 – CONCLUSÕES

Os objetivos deste trabalho consistiram na construção de um módulo de interação para um *framework* voltado ao treinamento médico, e a posterior avaliação de aplicações geradas por meio deste *framework*.

Os objetivos propostos inicialmente foram alcançados, visto que o módulo de interação com suporte a diversos dispositivos, tanto convencionais como não convencionais, foi construído utilizando-se das linguagens de programação Java e C++, juntamente com a API Java3D e incorporado ao *framework* ViMeT. A integração das linguagens de programação C++ e Java teve importante papel no desenvolvimento do projeto, gerando várias dificuldades, solucionadas durante o decorrer do projeto.

### 6.1 Contribuição para a Área de Computação

O trabalho apresentado propicia diversas contribuições para a área de Computação, citados a seguir. O primeiro ponto a ser destacado é que a integração entre linguagens de programação ainda é pouco detalhada na literatura existente; com a implementação e a documentação apresentadas, é possível auxiliar no desenvolvimento de aplicações desta natureza, inclusive para outros Sistemas Operacionais, aproveitando a portabilidade da linguagem de programação Java, e reduzindo os custos de desenvolvimento. Observa-se que o processo de integração de linguagens de programação pode ser adotado para desenvolver aplicações em outros domínios, não somente em benefício à Medicina, utilizando como mencionado, outras plataformas. Um segundo ponto a ser enfatizado é que as classes do ViMeT escritas em Java, que representam o módulo de interação podem ser usadas em outras

aplicações, para promover a utilização dos dispositivos (luva de dados, teclado, *mouse* e equipamento háptico).

Outro ponto importante a destacar é a possibilidade de escolha de dispositivos de acordo com a disponibilidade financeira. As aplicações podem usar dispositivos convencionais, de custo mais baixo, entretanto, não projetados para tarefas de forma natural nos AVs, como o teclado na manipulação dos dedos da mão virtual; e não convencionais, de custos mais elevados quando comparados aos convencionais, e projetados para tornar as tarefas mais próximas da realidade, como a luva de dados na movimentação dos dedos virtuais em concordância com a flexão dos dedos do usuário.

As características e funcionalidades do módulo de interação, bem como o ViMeT como um todo, podem ser usadas na construção de aplicações para outras áreas que fazem uso da computação, e necessitam por exemplo, de um método de detecção de colisão, ou de uma forma de importação de objetos tridimensionais, ou de interagir com AVs por meio de dispositivos convencionais e não convencionais, como os que compõem o módulo de interação. Essas características e funcionalidades podem ser modificadas para atender outras necessidades, de outras áreas do conhecimento, além da possibilidade da adição de novas características e funcionalidades de forma fácil.

## **6.2 Contribuição para a Área Médica**

Para a área médica, o ViMeT propicia a geração de aplicações para treinamento de exames de biópsia, utilizando dispositivos convencionais e não convencionais, oferecendo opções para inclusão de dispositivos de acordo com a disponibilidade financeira e o grau de realismo desejado.

O treinamento não está completamente implementado; entretanto, diversas etapas foram planejadas e construídas de forma que podem ser modificadas de acordo com a necessidade, proporcionando flexibilidade e facilitando o aperfeiçoamento do sistema, e conseqüentemente, da qualidade de interação, pois aplicações médicas exigem precisão e respostas em tempo real. A estrutura do ViMeT permite também que novas funcionalidades sejam adicionadas, inclusive outros dispositivos, métodos de colisão e deformação, além de outros pontos de vista para visualização.

A estrutura atual propicia que aplicações para simular outros procedimentos médicos possam ser geradas com as devidas adaptações. Os sistemas de RV apresentam diversos outros benefícios, além da diminuição ou eliminação de riscos no treinamento, como o fato do AV para simulação médica estar sempre disponível, a capacidade de simular diversos casos (diferentes anomalias, por exemplo) e possibilidade de incluir novos casos, descobertas da Medicina (novas técnicas, por exemplo), a minimização do uso de cobaias, a possibilidade do profissional da área verificar seu desempenho em todas as etapas de um determinado procedimento médico, ou visualizá-lo considerando perspectivas distintas para um melhor entendimento.

Desta forma, a aquisição de habilidade para executar certos procedimentos pode ser facilitada, bem como a aquisição de experiência, tornando o usuário especialista na realização dos procedimentos e aumentando a confiabilidade. No entanto, o caminho a ser percorrido na construção de simuladores condizentes com a realidade é desafiador, constituindo diferentes vertentes de pesquisa.

### 6.3 Trabalhos Futuros

A avaliação do primeiro protótipo foi determinante para traçar as próximas etapas da construção do ViMeT. No que diz respeito a trabalhos futuros, diversas questões poderão ser tratadas:

- construção de modelos tridimensionais com texturas mais próximas da realidade;
- implementação de métodos mais precisos para detecção de colisão e deformação, levando em consideração as diferentes camadas que compõem o órgão humano;
- métodos mais precisos para o retorno de força, considerando as diversas camadas do órgão humano, o ângulo do instrumento médico no início, durante a perfuração da pele e no momento da extração;
- a adoção de um detector de posição para a luva de dados, podendo desta forma, captar os movimentos da mão do usuário;
- correlação espacial entre os dispositivos e os objetos virtuais correspondentes;
- inclusão de pontos de vista adicionais no AV;
- finalização do procedimento de biópsia;
- avaliação das aplicações com uma amostra maior de usuários para definição de avaliação estatística mais significativa;
- confecção de questionários mais elaborados, levando em consideração o grau de realismo necessário às aplicações. Os trabalhos futuros representam melhorias a serem realizadas no ViMeT e, conseqüentemente, nas aplicações geradas, bem como a implementação de outras partes para que um treinamento completo de exame virtual de biópsia possa ser executado.

## 6.4 Limitações

Algumas limitações da aplicação podem ser citadas, tais como:

- em alguns casos, a rotação da seringa ocorre de forma incorreta;
- algumas vezes, o retorno de força torna-se contínuo mesmo não havendo colisão entre os objetos virtuais;
- em alguns casos, há erros de execução na colisão de objetos quando o objeto ou os objetos rígidos são os dedos da mão virtual, tocando o objeto deformável que representa o órgão humano.

Com relação à rotação da seringa, diversos testes foram realizados; no entanto, uma solução satisfatória não foi encontrada, uma vez que o problema é intermitente. O mesmo ocorreu com o retorno de força, que em alguns momentos é cessado, e em outros ocorre de forma contínua, mesmo não havendo colisão entre o objeto que representa o instrumento médico e o objeto que representa o órgão humano.

Tais limitações poderão ser incluídas em trabalhos futuros para análise e implementação de possíveis soluções, objetivando a construção de aplicações mais consistentes e próximas da realidade.

## 6.5 Trabalhos Publicados

Os resultados obtidos durante a realização deste projeto geraram algumas publicações, citadas nas referências bibliográficas desta dissertação: Corrêa *et al.* (2007a), Corrêa *et al.* (2007b), Corrêa *et al.* (2008a), Corrêa *et al.* (2008b), Corrêa *et al.* (2009).

## REFERÊNCIAS

*5DT Fifth Dimension Technologies*. Disponível em: <<http://www.5dt.com/hardware.html>> Acesso em: 06 Mar. 2007.

*Absolute Technologies*. Disponível em: <<http://www.abs-tech.com>>. Acesso em: 27 abr. 2007.1

BAECKER, R.; CARD, S.; CAREY, T.; GASEN, J.; HEWETT, T. T.; MANTEI, M.; PERLMAN, G.; STRONG, G.; VERPLANK, W. 2004. *Proceedings of ACM SIGCHI Curricula for Human-Computer Interaction*. Disponível em: <<http://sigchi.org/cdg/cdg2.html>>. Acesso em: 07 ago. 2006.

BANGAY, S.; RORKE, M.; WENTWORTH, P. *Virtual Reality Interaction Techniques*. 1998. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/2588/http://zSzzSzwww.cs.ru.ac.zazSzvrsigzSztechdocszSzrorkezSzsatnac-paper.pdf/rorke98virtual.pdf>> Acesso em: 10 dez. 2006.

BARCIA, R. M.; LUZ, R. P. da; MERINO, E.; REBELO, I. B. *Evaluation of VR System: More Usable Interactions*. Disponível em: <[http://citeseer.ist.psu.edu/cache/papers/cs/30416/http://zSzzSzwww.lrv.ufsc.brzSz.zSzrecursozSzartigoszSzEvaluation\\_VR\\_Systems.pdf/evaluation-of-vr-systems.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/30416/http://zSzzSzwww.lrv.ufsc.brzSz.zSzrecursozSzartigoszSzEvaluation_VR_Systems.pdf/evaluation-of-vr-systems.pdf)> Acesso em: 12 fev. 2007.

BASTOS, T. A.; RAPOSO, A. B.; GATTAS M. Um *Framework* para o Desenvolvimento de Aplicações de Realidade Virtual Baseados em Componentes Gráficos. In: XXV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, São Leopoldo, 2005. **Anais do XXV Congresso da Sociedade Brasileira de Computação**. São Leopoldo, 2005. p. 213-223.

BATTEAU, L. M.; BHASIN, Y.; BOWYER M. W.; LIU, A.; MAINTZ, J. B. A. *A Study on the Perception of Haptics in Surgical Simulation. Lecture Notes in Computer Science – LNCS, Medical Simulation*. Springer Berlin, Heidelberg, Alemanha, Volume 3078, p. 185-192, jun. 2004.

BERALDIN, J.-A.; BOULANGER, P.; EL-HAKIM, S. F.; GODIN, G. *Two 3-D Sensors for Environment Modeling and Virtual Reality: Calibration and Multi-View Registration*. In: *International Achieves of Photogrammetry and Remote Sensing*, Volume 31, Parte B5, Viena, Áustria, 1996. *Proceedings of International Achieves of Photogrammetry and Remote Sensing*. Viena, Áustria, 1995. p.140-146.

BERNIER, F.; BRANZAN-ALBU A.; DROUIN, R.; LAURENDEAU, D.; LIMIEUX, F.; MOKHTARI, M.; QUELLET, D. ***Virtual Environment and Sensori-Motor Activities: Visualization***. In: *WSCG – 12th International Conference in Central Europe Computer Graphics, Visualization and Computer Vision*, Plzen-Bory, República Tcheca, 2004. ***Proceedings of 12th International Conference in Central Europe Computer Graphics, Visualization and Computer Vision***, Plzen-Bory, República Tcheca, 2004. p.102-112.

BEZERRA, A.; NUNES, F. L. S.; CORRÊA, C. G. ***Interação com Equipamentos Convencionais e Não Convencionais em Treinamento Médico***. In: *SVR - X Symposium on Virtual and Augmented Reality*, João Pessoa, PB, Brasil, 2008. ***Proceedings of the X Symposium on Virtual and Augmented Reality***, João Pessoa, PB, Brasil, 2008. p. 275-278.

BOSCH, J.; MATSSON, M. ***Framework Composition: Problems, Causes and Solutions***. In: *TOOLS – International Conference on Technology of Object Oriented Systems and Languages*, Santa Barbara, CA, USA, 1997. ***Proceedings of TOOLS – International Conference on Technology of Object Oriented Systems and Languages***. Santa Barbara, CA, USA, 1997.p.203-214. IEEE-Computer Society.

BOTEGA, L. C.. ***Implementação de Estereoscopia de Baixo Custo para Aplicações em Ferramentas de Realidade Virtual para Treinamento Médico***. 2005. 105 f. Grau: Monografia (Bacharelado em Ciência da Computação) Centro Universitário Eurípides de Marília – Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

BOWMAN, D. A.; GABBARD, J.; HIX, D. ***Usability Evaluation in Virtual Environments: Classification and Comparison of Methods***. *Computer Science*, 2001 (a). Disponível em: <[eprints.cs.vt.edu/archive/00000541/01/VE\\_usability\\_presence.pdf](http://eprints.cs.vt.edu/archive/00000541/01/VE_usability_presence.pdf)> Acesso em: 10 dez. 2006. ACM Press

BOWMAN, D. A.; HODGES, L. F. ***An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments***. In: *Symposium on Interactive 3D Graphics*, Providence, Rhode Island, USA, 1997. ***Proceedings of the Symposium on Interactive 3D Graphics***, Providence, Rhode Island, USA, 1997. p.5-38. ACM

BOWMAN, D. A.; HODGES, L. F.; JOHNSON, D. B. ***Testbed Evaluation of Virtual Environment Interaction Techniques***. In: *Symposium on Virtual Reality Software and Technology*, Londres, Reino Unido, 1999. ***Proceedings of the ACM Symposium on Virtual Reality Software and Technology***, Londres, Reino Unido, 1999. p. 26-33.



BOWMAN, D. A.; HODGES, L. F.; KOLLER, D. A. *A Methodology for the Evaluation of Travel Techniques for Immersive Virtual Environments*. **Journal of Virtual Reality, Computer Science**. Springer London, Londres, Reino Unido, Volume 3, Número 2, p. 120-131, jun. 1998.

BOWMAN, D. A.; JR, J. J. L.; KRUIJFF, E.; POUPYREV, I. *An Introduction to 3-D User Interface Design. Presence: Teleoperators and Virtual Environments*, Vol. 10, Número 1, 2001(b), p. 96-108.

BREWSTER, S.; GRAY, P.; MCGEE, M.; OAKLEY, I. *Putting the Feel in Look and Feel*. In: CHI – *Conference on Human Factors in Computing Systems*, Hague, Holanda, 2000. *Proceedings of the Conference on Human Factors in Computing Systems*, Hague, Holanda, 2000. p. 415-422. ACM Press

BROCKWAY, D.; BURNETTE, T.; PAUSCH, R.; WEIBLEN, M. E. *Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures*. In: *22<sup>nd</sup> Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 1995. *Proceedings of the 22<sup>nd</sup> Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 1995. p.399-400.

BURDEA, G. C. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons, Inc.: New York, USA, 1996.

BURDEA, G.; COIFFET, P. *Virtual Reality Technology*. John Wiley & Sons, Inc.: New York, USA, 1994. p. 02.

BURNS, J. M.; CHILA, A. G.; ELAND, D. C.; HOWELL, J. N.; JR, R. R. C.; SRIVASTAVA, M.; WILLIAMS, R. L. *The Virtual Haptic Back for Palpatory Training*. In: *6<sup>th</sup> International Conference on Multimodal Interfaces*, State College, PA, USA, 2004. *Proceedings of the 6th International Conference on Multimodal Interfaces*, State College, PA, USA, 2004. p. 191-197.

CAMARERO, R.; OZELL, B.; PIC, C. *Topics in Collaborative Visualization Environments*. 2000. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/20998/http:zSzzSzwww.cerca.umontreal.ca:zSzeur:zSzScience:zSzRapport:zSzFichiersps:zSzR2000-84.pdf/ozell00topics.pdf>> Acesso em: 12 fev. 2007.

CELES, W.; GATTASS, M.; RAPOSO, A. B.; SZENBERG F. **Visão Estereoscópica, Realidade Virtual, Realidade Aumentada e Colaboração**. Disponível em: <<http://www.tecgraf.puc->

rio.br/publications/artigo\_2004\_visao\_estereoscopica\_realidade\_virtual.pdf> Pontifícia Universidade Católica do Rio, Rio de Janeiro, 2004. Acesso em: 27 abr. 2007.

CHUI, Y.; HENG, P.; LEUNG, K.; SUN, H.; WONG, T. **A Haptic Needle Manipulation Simulator for Chinese Acupuncture Learning and Training**. In: *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality*. Disponível em: <<http://portal.acm.org/citation.cfm?id=1044598&coll=ACM&dl=ACM&CFID=19790155&CFTOKEN=97492316>> 2004. Acesso em: 15 out. 2006.

COELHO, Y. da S.; PEPPE, M. J. **A Realidade Virtual na Medicina**. Disponível em: <<http://www.eca.usp.br/nucleos/njr/voxscientiae/caverna7.html>>. *Vox Scientiae*, São Paulo, 2002. Acesso em: 09 jun. 2006.

CONWAY, M.; HINCKLEY, K.; KASSELL, N. F.; PAUSCH, R.; PROFFITT, D.; STOAKLEY, R. **Revisiting Haptic Issues for Virtual Manipulation**. Universidade da Virginia, Estados Unidos. Disponível em: <<http://www.cs.virginia.edu/papers/manip.pdf>> Acesso em: 10 dez. 2006.

CORNELL, G.; HORSTMANN, C. S. **Métodos Nativos**. Core Java 2, Volume 2, Recursos Avançados. Pearson Education do Brasil, São Paulo, SP, Brasil, 2003. pp. 755-785.

CORRÊA, C. G.; BEZERRA, A.; NUNES, F. L. S. **Implementação de Realismo na Interação em um Framework para Treinamento Médico**. In: SIIM – Simpósio de Imagens e Instrumentação Médica, São Carlos, SP, Brasil, 2007. CD-ROM (a)

CORRÊA, C. G.; BEZERRA, A.; OLIVEIRA, A. C. M. T. G.; NUNES, F. L. S. **Implementação de Realismo na Interação em um Framework para Treinamento Médico**. In: WRVA – Workshop de Realidade Virtual e Aumentada, Itumbiara, GO, Brasil, 2007. CD-ROM (b)

CORRÊA, C. G.; BEZERRA, A.; NUNES, F. L. S. **Integração de Linguagens de Programação para Uso de Dispositivos Não convencionais: Possível Solução para Construir Aplicações com Baixo Custo**. In: *X Symposium on Virtual and Augmented Reality*, João Pessoa, PB, Brasil, 2008. *Proceedings of the X Symposium on Virtual and Augmented Reality*, João Pessoa, PB, Brasil, 2008. p. 266-269. (a)

CORRÊA, C. G.; BEZERRA, A.; NUNES, F. L. S. **Implementação de Interação em Sistemas Virtuais para Simulação de Exames de Biópsia**. In: Workshop de Informática Médica, Belém, PA, Brasil, 2008. *Proceedings of the Workshop de Informática Médica*, CD-ROM, Belém, PA, Brasil, 2008. (b)

CORRÊA, C. G.; BEZERRA, A.; NUNES, F. L. S.; CARVALHO JR., P. M. **Evaluation of VR Medical Training Applications under the Focus of Professionals of the Health Area.** In: 24<sup>th</sup> SAC – *International Symposium on Applied Computer*, Hawaii, U.S.A., 2009. **Proceedings of the 24<sup>th</sup> International Symposium on Applied Computer**, Hawaii, U.S.A., 2009. ACM.

DEITEL, H. M.; DEITEL, P. J. **Java How to Program.** Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

EISENSTEIN, J.; GHANDEHARIZADEH, S.; GOLUBCHIK, L.; SAHABI, C.; YAN, D.; ZIMMERMANN, R. **Device Independence and Extensibility in Gesture Recognition.** In: *IEEE Virtual Reality*, 2003. **Proceedings of the IEEE Virtual Reality**, 2003. p. 207-214.

EST – *Engineering Systems Technologies*. Disponível em: <[http://www.est-kl.com/pricelist/pricelist\\_gb.html](http://www.est-kl.com/pricelist/pricelist_gb.html)>. Acesso em: 13 Mar. 2007.

FERRO, E. N. de L.; PAIVA, A. C. de. **Modelagem e Implementação de um Sistema para Treinamento de Técnicos em Radiologia Baseado em Realidade Virtual Não Imersiva.** In: *VI SVR – Synposium on Virtual Reality*, Ribeirão Preto, São Paulo, Brasil, 2003. **Proceedings of the VI SVR – Symposium on Virtual Reality**, Ribeirão Preto, 2003. p.180-186.

FLASAR, J. **3D Interaction in Virtual Environment.** In: *4<sup>th</sup> Central European Seminar on Computer Graphics*, Budmerice, Eslováquia, 2000. **Proceedings of the 4<sup>th</sup> Central European Seminar on Computer Graphics**, Budmerice, Eslováquia, 2000. p.21-31.

FORSBERG, A.; HERNDON, K.; ZELEZNIK, R. **Aperture Based Selection from Immersive Virtual Environments.** In: *9<sup>th</sup> Annual ACM Symposium on User Interface Software and Technology*, Seattle, Washington, USA, 1996. **Proceedings of the 9<sup>th</sup> Annual ACM Symposium on User Interface Software and Technology**, Seattle, Washington, USA, 1996. p.95-96.

FREITAS, C. M. D. S.; JACOB, L. J.; NEDEL, L. P.; OLIVA, C. B. **Avaliação Experimental de Técnicas de Seleção e Manipulação em Ambientes Virtuais.** In: *V SVR – Symposium on Virtual Reality*, Fortaleza, CE, 2002. **Proceedings of the V SVR – Symposium on Virtual Reality**, Fortaleza, CE, 2002. p. 38-49.

FREITAS, C. M. D. S.; NAVARRE, D.; NEDEL, L. P.; PALANQUE, P.; SCHYN, A. **Usando Modelagem Formal para Especificar Interação em Ambientes Virtuais: Por que?** In: *SVR – Synposium on Virtual Reality*, Ribeirão Preto, São Paulo, Brasil, 2003. **Proceedings of the VI SVR – Symposium on Virtual Reality**, Ribeirão Preto, 2003. p.81-92.

FREITAS JR, R. **Punção Aspirativa por Agulha Fina: Estudo Comparativo entre dois Dispositivos para a Obtenção da Amostra Citológica**, 2001, 136 f., Grau: Tese de Doutorado em Tocoginecologia, Faculdade de Ciências Médicas da Universidade Estadual de Campinas, Campinas, Brasil, Disponível em: <<http://www.rau-tu.unicamp.br/nou-rau/demopt/document/?view=57>>, Acesso em: 5 fev. 2008.

GARCIA, M. de B.; GONÇALVES, A. S.; JR, R. D. de S.; MEIGUINS, B. S.; SANTOS, D. N. A. **Interação em Ambientes Virtuais Tridimensionais Utilizando Comandos de Voz**. In: *SVR – Symposium on Virtual Reality*, Ribeirão Preto, São Paulo, Brasil, 2003. *Proceedings of the VI SVR – Symposium on Virtual Reality*, Ribeirão Preto, 2003. p.244-253..

FUKATSU, S.; KISHINO, F.; KITAMURA, Y.; MASAKI, T. **Intuitive Navigation in an Enormous Virtual Environment**. In: *ICAT – International Conference a Artificial Reality and Tele-Existence*, 1998. Disponível em: <<http://www-human.ist.osaka-u.ac.jp/IMCP/papers/icat98.pdf>> Acesso em: 10 dez. 2006. p. 163-169.

GABBARD, J. L.; HIX, D. **Usability Engineering of Virtual Environments**. Livro: STANNEY, K. *Handbook of Virtual Environments: Design, Implementation and Applications*, Lawrence Erlbaum Associates, 2002. p. 681-699.

GRANTZ, V.; ROESSLER, A. **Performance Evaluation of Input Devices in Virtual Environments**. In: *IFIP - Conference on Designing Effective and Usable Multimedia Systems*, Deventer, Holanda, Volume 133, 1998. *Proceedings of the IFIP - Conference on Designing Effective and Usable Multimedia Systems*, Deventer, Holanda, Volume 133, 1998. p. 68.

HADWIGER, M. **PARSEC: Enhancing Realism f Real-Time Graphics Through Multiple Layer Rendering and Particle System**. In: *CESCG – 2nd Central European Seminar on Computer Graphics*, Budmerice, Eslováquia, 1998. *Proceedings of the CESCG – 2nd Central European Seminar on Computer Graphics*, Budmerice, Eslováquia, 1998. p. 155-164.

HANSEN, C. D.; JOHNSON, C. R.; IKITS. M. **A Comprehensive Calibration and Registration Procedure for the Visual Haptic Workbench**. In: *Workshop on Virtual Environments*, Volume 39, Zurique, Suíça, 2003. *Proceedings of the Workshop on Virtual Environments*, Zurique, Suíça, 2003. ACM Press p. 247-254.

HSU, J. **Active Interaction Devices**. Disponível em: <<http://www.hitl.washington.edu/scivw/EVE/I.D.1.a.ActiveInteraction.html>> Acesso em: 10 dez. 2006.

HUFF, R.; SILVA, I. C. S. da; VASCONCELOS, A. B. **Seleção de Objetos em Ambientes Virtuais com Mouse 3D**. In: VIII SVR – *Symposium on Virtual Reality*, Belém, PA, Brasil, 2006. Anais do VIII SVR – *Symposium on Virtual Reality*, Belém, PA, Brasil, 2006. CD-ROM.

IMMERSION (2008). *CathSim AccuTouch System – Success Story*. Disponível em: <http://www.immersion.com>. Acesso em: 26 ago. 2008.

JAFRY, H; JOHNSON, K. W.; OKAMURA, A. M.; WEBSTER III, R. J. *The Haptic Scissors: Cutting in Virtual Environments*. In: *ICRA – IEEE Conference on Robotics and Automation*, Volume 1, Taipei, Taiwan, 2003. *Proceedings of the ICRA – IEEE Conference on Robotics and Automation*, Volume 1, Taipei, Taiwan, 2003. p. 828-833.

JAVA3D. Java3D API. Disponível em: <<http://java.sun.com/products/java-media/3D>> Acesso em: 20 jun. 2008.

KAUR, K.; MAIDEN, N.; SUTCLIFFE, A. *Improving Interaction with Virtual Environments*. In: *IEEE Colloquium on the 3D Interface for the Information Worker*, Londres, Reino Unido, 1998. *Proceedings of the IEEE Colloquium on the 3D Interface for the Information Worker*, Londres, Reino Unido, 1998. p.04.

KENNEDY, R. S.; MOURANT, R. R.; STANNEY, K. M. *Human Factors Issues in Virtual Environments: A Review of the Literature*. Volume 7, Número 4, *Massachusetts Institute of Technology* (MIT), 1998. Disponível em: <<http://web.mit.edu/16.459/www/Stanne.pdf> > Acesso em: 10 dez. 2006. p. 327-351.

KERA, Marcello. **Detecção de colisão utilizando hierarquias em ferramentas de realidade virtual para treinamento médico**. 2005 92 f. Grau: Monografia (Bacharelado em Computação) Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005. Disponível em: < <http://lost.fundanet.br/~kera>>. Acesso em 20 dez. 2005

KIM, Y. J.; LIN, M. C.; MANOCHA, D.; OTADUY, M. A. *Six-Degree-of-Freedom Haptic Display Using Localized Contact Computations*. In: *10<sup>th</sup> Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2002. *Proceedings of the 10<sup>th</sup> Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2002. p. 209. IEEE – Computer Society.

KIRNER, C. **Sistemas de Realidade Virtual**. Universidade Federal de São Carlos – UFSCar, São Carlos, SP, Brasil. Disponível em: <<http://www.dc.ufscar.br/~grv/tutrv/tutrv.htm>> Acesso em: 10 dez. 2006.

KOIKE, H.; SAITO, M.; SATO, Y. **Real-Time Input of 3D Pose and Gestures of a User's Hand and Its Applications for HCI**. 2001. *Proceedings of the IEEE on Virtual Reality*, 2001. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/25082/http:zSzzSzjpn.hci.iis.u-tokyo.ac.jpzSz~ysatozSzpaperszSzVR2001.pdf/real-time-input-of.pdf>> Acesso em: 12 fev. 2007. p.79-86.

KOPPER, R. A. P.; PINHO, M. S.; RIEDER, R.; SANTOS, M. C. C. dos; SILVA, F. B. de A. e; TROMBETTA, A. B. **Uma Avaliação Sobre o Uso de Estímulos Táteis em um Ambiente Virtual**. In: *VIII SVR Symposium on Virtual Reality*, Belém, PA, Brasil, 2006. CD-ROM.

KORNER, O.; MANNER, R. **Haptic Display for a Virtual Reality Simulator for Flexible Endoscopy**. In: *ACM International Conference Proceeding Series*, Volume 2, 2002. *Proceedings of the workshop on Virtual environments*, Barcelona, Espanha, 2002. p. 13.

LEIBE, B.; PAIR, J.; SINGLETARY, B.; STARNER, T. **MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Game**. In: *5<sup>th</sup> International Conference on Intelligent User Interfaces*, New Orleans, Louisiana, United States, 2000. *Proceedings of the 5<sup>th</sup> International Conference on Intelligent User Interfaces*, New Orleans, Louisiana, United States, 2000. ACM Press p. 256-259.

LOPES, R. de D.; MACHADO, L. dos S.; MORAES, R. M.; ZUFFO, M. K. **Modelagem Tátil, Visualização Estereoscópica e Aspectos de Avaliação em um Simulador de Coleta de Medula Óssea**. In: *IV SBC Symposium on Virtual Reality*, Florianópolis, SC, Brasil, 2001. *Anais do IV SBC Symposium on Virtual Reality*, Florianópolis, SC, Brasil, 2001. p. 26-28.

LSI-USP - Laboratório de Sistemas Integráveis – Universidade de São Paulo. Disponível em: <<http://www.lsi.usp.br/interativos/nrv/caverna.html>>. Acesso em: 13 Mar. 2007.

MACHADO, L. dos S. **A Realidade Virtual no Modelamento e Simulação de Procedimentos Invasivos em Oncologia Pediátrica: Um Estudo de Caso No Transplante de Medula Óssea**. Tese de Doutorado, Universidade de São Paulo, São Paulo, 2003. (a)

MACHADO, L. dos S.; NETTO, A. V.; OLIVEIRA, M. C. F. de. **Realidade Virtual – Fundamentos e Aplicações**. Visual Books Editora: Florianópolis, 2002. pp 38-40. (b)

MACHADO, L. dos S. **Aplicações da Realidade Virtual em Cirurgias**. Livro do VII Symposium on Virtual Reality: Realidade Virtual – Conceitos e Tendências. Editora Mania de Livro: São Paulo, 2004. p. 271 (c)

MACHADO, L. dos S.; NETTO, A. V.; OLIVEIRA, M. C. F. de. **Realidade Virtual – Definições, Dispositivos e Aplicações**. Notas Didáticas, Número 34, São Carlos, SP, Brasil, 2002. Disponível em: <<http://www.sbc.org.br/reic/edicoes/2002e1/tutoriais/RV-DefinicoesDispositivosEAplicacoes.pdf>>. 2002 (d). Acesso em: 12 abr. 2007. p. 33.

MACHADO, L. S.; MORAES, R. M.; SOUZA, D. F. L.; VALDEK, M. C. O. **SITEG – Sistema Interativo de Treinamento em Exame Ginecológico**. In: *VI SVR – Synposium on Virtual Reality*, Ribeirão Preto, São Paulo, Brasil, 2003. CD-ROM.

MATTSSON, M. **Object-Oriented Frameworks: A survey of methodological issues**. In: *IFAC – 13<sup>th</sup> International Federation of Automatic Control World Congress*, São Francisco, CA, USA, 1996. *Proceedings of the 13<sup>th</sup> International Federation of Automatic Control World Congress*, São Francisco, CA, USA, 1996. p. 130.

Microsoft Inc. (2008) Disponível em: <<http://www.microsoft.com.br/brasil/msdn/cplusplus/Default.aspx>> Acesso em: 31 mar. 2008.

MINE, M. R. **Virtual Environment Interaction Techniques**. 1995. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/1796/ftp:zSzzSzftp.cs.unc.edu/SzpubzSztechnical-reportszSz95-018.pdf/mine95virtual.pdf>> Acesso em: 10 dez. 2006.

MONTERO, E. F. de S.; ZANCHET, D. J. **Realidade Virtual e a Medicina**. In: *Scientific Electronic Library Online*, Acta Cirúrgica Brasileira, Vol.18, No.5, São Paulo, SP, Brasil 2003. p. 489-490.

NUNES, F. L. S.; COSTA, R. M. E. M.; OLIVEIRA, A. C. M. T. G.; DELFINO, S. R.; PAVARINI, L.; RODELLO, I. A.; BREGA, J. R. F.; SEMENTILLE, A. C. **Aplicações Médicas Usando Realidade Virtual e Realidade Aumentada**. In: C. Kirner, R. Siscoutto, *Realidade Virtual e Aumentada – Conceitos, Projeto e Aplicações*, Livro do 9º SVR *Synposium on Virtual and Augumented Reality*, Petrópolis, RJ, Brasil, 2007. p.234-235. (a)

NUNES, F. L. S.; OLIVEIRA, A. C. M. T. G.; ROSSATO, D. J.; MACHADO, M. I. C. **ViMeTWizard: Uma ferramenta para instanciação de um framework de Realidade Virtual para treinamento médico**. In: XXXIII Conferencia Latinoamericana de Informática, San José, 2007. *Proceedings of XXXIII Conferencia Latinoamericana de Informática*, San José, 2007. p. 1 – 8. (b)

OLIVEIRA, A. C. M. T. G. de. **ViMeT - Um Framework em Java para Aplicações de Treinamento Médico usando Realidade Virtual.** In: IX SVR - *Symposium on Virtual Reality*, Petrópolis, RJ, 2007. *Proceedings of the Symposium on Virtual Reality*, Petrópolis, RJ, 2007. p. 253-262.

OLIVEIRA, A. C. M. T. G.; DELFINO, S. R.; NUNES, F. L. S. **Virtual Reality Framework for Medical Training: Implementation of a deformation class using Java.** In: *Proceedings of the SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, Hong Kong, China, 2006. *Proceedings of the SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, Hong Kong, China, 2006. ACM Press, New York, NY, 2006. p. 347-351.

OOMS, M. **Measuring presence and awareness of external events.** Twente University, Holanda, 2004. Disponível em: <[http://citeseer.ist.psu.edu/cache/papers/cs2/94/http://zSzzSzreferaat.ewi.utwente.nlzSzdOCUMENTS2/2004\\_01\\_C-Intelligent\\_InteractionzS2004\\_01\\_C\\_M.J.Ooms-Measuring\\_presence\\_and\\_awareness\\_of\\_external\\_events.pdf/measuring-presence-and-awareness.pdf](http://citeseer.ist.psu.edu/cache/papers/cs2/94/http://zSzzSzreferaat.ewi.utwente.nlzSzdOCUMENTS2/2004_01_C-Intelligent_InteractionzS2004_01_C_M.J.Ooms-Measuring_presence_and_awareness_of_external_events.pdf/measuring-presence-and-awareness.pdf)> Acesso em: 19 fev. 2007.

PAVARINI, L. **Estudo e Implementação do Método massa-mola para Deformação em Ambientes Virtuais de Treinamento Médico usando a API Java 3D.** 2006. 147f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

PINHANEZ, C. **Interfaces Não convencionais.** In: Livro do VII *Symposium on Virtual Reality: Realidade Virtual – Conceitos e Tendências*. Editora Mania de Livro: São Paulo, 2004. p. 135

PINHO, M. S. **Interação em Ambientes Tridimensionais.** Minicurso. In: WORKSHOP DE REALIDADE VIRTUAL, Gramado, RS, Brasil, 2000. Disponível em: <<http://www.inf.pucrs.br/~pinho/3DInteraction/>>. Acesso em: 10 out. 2006.

PINHO, M. S.; REBELO, I. B. **Interação em Ambientes Virtuais Imersivos.** In: Livro do VII *Symposium on Virtual Reality: Realidade Virtual – Conceitos e Tendências*. Editora Mania de Livro: São Paulo, 2004. p. 109

PINHO, M. S.; SILVA, F. B. de A. **Avaliação de Técnicas de Auxílio a Wayfinding em Ambientes Virtuais.** In: IX *Symposium on Virtual and Augmented Reality*. Petrópolis, RJ, Brasil, 2007. *Proceedings of the IX Symposium on Virtual and Augmented Reality*, Petrópolis, RJ, Brasil, 2007. pp. 143-151.



RIVA, G. *Applications of Virtual Environments in Medicine*. MIM – *Methods of Information in Medicine*. Istituto Auxologico Italiano, Milan, Itália, Volume 42, Número 05, 2003. Disponível em: <[http://www.cybertherapy.info/MIM\\_0161\\_Riva.pdf](http://www.cybertherapy.info/MIM_0161_Riva.pdf)> Acesso em: 10 dez. 2006. p. 524-534.

ROCHA, L. A. **Formas de Interação entre Humanos e Dados Digitais em Ambientes Virtuais**. In: Horizontes Antropológicos, Ano 10, Número 21, UFRGS (Universidade Federal do Rio Grande do Sul), Porto Alegre, RS, Brasil, 2004. p. 158.

SATALICH, G. A. *Navigation and Wayfinding in Virtual Reality: Finding Proper Tools and Cues to Enhance Navigation Awareness*. Disponível em: <<http://www.hitl.washington.edu/publications/satalich/ref.html>> Acesso em: 15 dez. 2006.

*SensAble Technologies*. Disponível em: <http://www.sensable.com>. Acesso em: 06 Mar. 2007.

SILVA, R. J. M. **Integração de um Dispositivo Óptico de Rastreamento a uma Ferramenta de Realidade Virtual**. Dissertação de Mestrado. PUC (Pontifícia Universidade Católica), Rio de Janeiro, RJ, Brasil, 2004. Disponível em: <[http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/PRG\\_0599.EXE/6664\\_3.PDF?NrOcoSis=18800&CdLinPrg=pt](http://www.maxwell.lambda.ele.puc-rio.br/cgi-bin/PRG_0599.EXE/6664_3.PDF?NrOcoSis=18800&CdLinPrg=pt)> Acesso em: 10 dez. 2006.

SUN. *Java Native Interface Specification*. Disponível em: <<http://java.sun.com/j2se/1.4.2/docs/guide/jni/spec/jniTOC.html>>. Acesso em: 15 mar. 2007.

THALMANN, D. *Using Virtual Reality Techniques*. Disponível em: <[http://citeseer.ist.psu.edu/cache/papers/cs/27653/http%3A%2F%2FzSzzSzvrlab.epfl.chzSzPublicationszSzpdfzSzThalmann\\_VRS\\_93.pdf/using-virtual-reality-techniques.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/27653/http%3A%2F%2FzSzzSzvrlab.epfl.chzSzPublicationszSzpdfzSzThalmann_VRS_93.pdf/using-virtual-reality-techniques.pdf)> Acesso em: 12 fev. 2007.

ZIEGELER, S. B. *Using Virtual Environments to Visualize Atmospheric Data: Can It Improve a Meteorologist's Potential to Analyze the Information?* 2002. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/26818/http%3A%2F%2FzSzzSzwww.erc.msstate.eduSzvailSzpubszSz2002zSzsean01.pdf/ziegeler02using.pdf>>. Acesso em: 12 fev. 2007.

## APÊNDICE A

### A.1 Questionário Pré-Teste

Nome: \_\_\_\_\_ Idade: \_\_\_\_\_

1 – Sobre Realidade Virtual, envolvendo a utilização de dispositivos, qual é o seu nível de conhecimento?

- Conheço o termo Realidade Virtual (definição, aplicações etc), e já utilizei dispositivos de Realidade Virtual
- Conheço o termo Realidade Virtual (definição, aplicações etc), e nunca utilizei dispositivos de Realidade Virtual
- Não sei o que significa Realidade Virtual

2 - Se assinalar a primeira alternativa da Questão 1, identifique os dispositivos que já usou:

- Dispositivo háptico
- Luva de dados
- Capacete
- Óculos estereoscópicos
- Outro(s): \_\_\_\_\_

3 – Sobre a execução de procedimentos de biópsia, qual é o seu nível de conhecimento?

- Sou da área médica (estudante de Medicina ou médico), e executo estes tipos de procedimentos
- Sou da área médica (estudante de Medicina ou médico), entretanto, ainda não executo estes tipos de procedimentos
- Não sou da área médica e não executo estes tipos de procedimentos

Se assinalar a primeira alternativa, identificar:

Tempo de experiência nestes tipos de procedimentos: \_\_\_\_\_

Média do número de procedimentos realizados: \_\_\_\_\_

4 – Na sua opinião, simuladores podem auxiliar no aprendizado de procedimentos médicos, como punção?

- Sim, simuladores podem ser considerados de grande importância no aprendizado de procedimentos médicos
- Sim, entretanto, precisam ser aperfeiçoados
- Ainda não podem auxiliar no aprendizado

## A.2 Questionário Pós-Teste

Nome: \_\_\_\_\_ Idade: \_\_\_\_\_

1 – Com relação aos dispositivos não convencionais (equipamento háptico e luva de dados), como se sentiu ao utilizá-los?

- Sem grandes dificuldades com a luva de dados, pois já participei de atividades com este equipamento ou adaptei-me de forma rápida a ele durante os testes
- Sem grandes dificuldades com o dispositivo háptico, pois já participei de atividades com este equipamento ou adaptei-me de forma rápida a ele durante os testes
- Senti algumas dificuldades para adaptar-me a luva de dados, mas não tive problemas com o dispositivo háptico
- Senti algumas dificuldades para adaptar-me ao dispositivo háptico, mas não tive problemas com a luva de dados
- Não me senti bem com nenhum dos dispositivos

2 – A tarefa a ser realizada no Ambiente Virtual foi explicada:

- De forma clara
- Parcialmente clara, deixando algumas dúvidas
- Não foi bem explicada

3 – Em uma escala de 1 a 4, defina os dispositivos de acordo com a facilidade de uso (o mais fácil deve receber o valor 1, e assim sucessivamente).

- Mouse*
- Teclado
- Luva de dados
- Dispositivo háptico

4 – No que diz respeito ao aspecto visual, pode-se dizer que a montagem do Ambiente Virtual (aparência e comportamento dos objetos virtuais):

- Teve importante papel na realização da simulação de treinamento
- Pouco contribuiu para a realização da simulação de treinamento
- Não contribuiu para a realização da simulação de treinamento

5 – No que diz respeito à utilização de dispositivos convencionais, pode-se dizer que:

- Tiveram importante papel na realização da simulação de treinamento
- Pouco contribuíram para a realização da simulação de treinamento
- Contribuíram satisfatoriamente para a realização da simulação de treinamento, quando foram combinados com os dispositivos não convencionais
- Contribuíram parcialmente para a realização da simulação de treinamento, quando foram combinados com os dispositivos não convencionais

Não contribuíram de forma nenhuma para a realização da simulação de treinamento

6 – No que diz respeito à utilização de dispositivos não convencionais, pode-se dizer que:

Tiveram importante papel na realização da simulação de treinamento

Pouco contribuíram para a realização da simulação de treinamento

Contribuíram satisfatoriamente para a realização da simulação de treinamento, quando foram combinados com os dispositivos convencionais

Contribuíram parcialmente para a realização da simulação de treinamento, quando foram combinados com os dispositivos convencionais

Não contribuíram de forma nenhuma para a realização da simulação de treinamento

7 – Sobre a combinação de dispositivos, em uma escala de 1 a 4, qual lhe agradou mais e contribuiu de forma satisfatória na realização da tarefa? A melhor combinação deve receber o valor 1, e assim sucessivamente, até o valor 4.

Dispositivo háptico e luva de dados

Dispositivo háptico e teclado

*Mouse* e luva de dados

*Mouse* e teclado

8 - Na sua opinião, simuladores podem auxiliar no aprendizado de procedimentos médicos, como exames de biópsia?

Sim, simuladores podem ser considerados de grande importância no aprendizado de procedimentos médicos

Sim, entretanto, precisam ser aperfeiçoados

Ainda não podem auxiliar no aprendizado

9 – Que tipo de procedimentos em Medicina podem ser beneficiados com simuladores virtuais?

10 – Se deseja apresentar sugestões, críticas, idéias, comentários, ou até mesmo, descrever sua experiência, por gentileza, utilize o espaço abaixo:

## APÊNDICE B

### B.1 Informações Técnicas de Configuração do Compilador

Para realizar a configuração do compilador *Microsoft Visual C++ 6.0*, permitindo o desenvolvimento de aplicações com funções existentes nas bibliotecas disponibilizadas pelo fabricante, há um roteiro no manual do dispositivo háptico, que será apresentado nesta seção com algumas adaptações.

No menu *Project*, escolhe-se a opção *Settings*, e na caixa que aparece, como pode-se observar na Figura 77, deve-se selecionar a guia *C/C++*, categoria *Code Generation*.

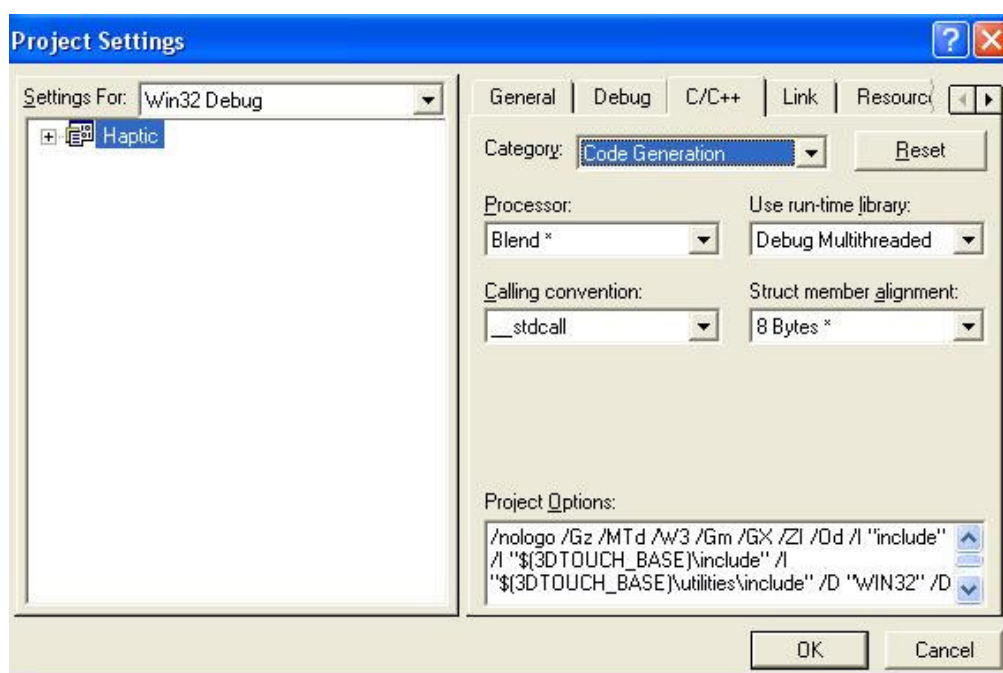


Figura 77 - Janela *Project Settings*

Para *Processor*, *Use run-time library*, *Calling convention* e *Struct member alignment*, deve-se selecionar as opções *Blend \**, *Debug Mutithreaded*, *\_stdcall*, *8 bytes \**, respectivamente, configurando assim os parâmetros de execução do código.

Nesta mesma janela, deve ser escolhida a categoria *Preprocessor*, e na caixa de texto *Additional include directories*, deve incluir a seguinte linha:

`include,$(3D TOUCH_BASE)\include,$(3D TOUCH_BASE)\utilities\include`, para indicar ao compilador onde estão localizadas as bibliotecas do *OpenHapticsToolkit*, conforme a Figura 78.

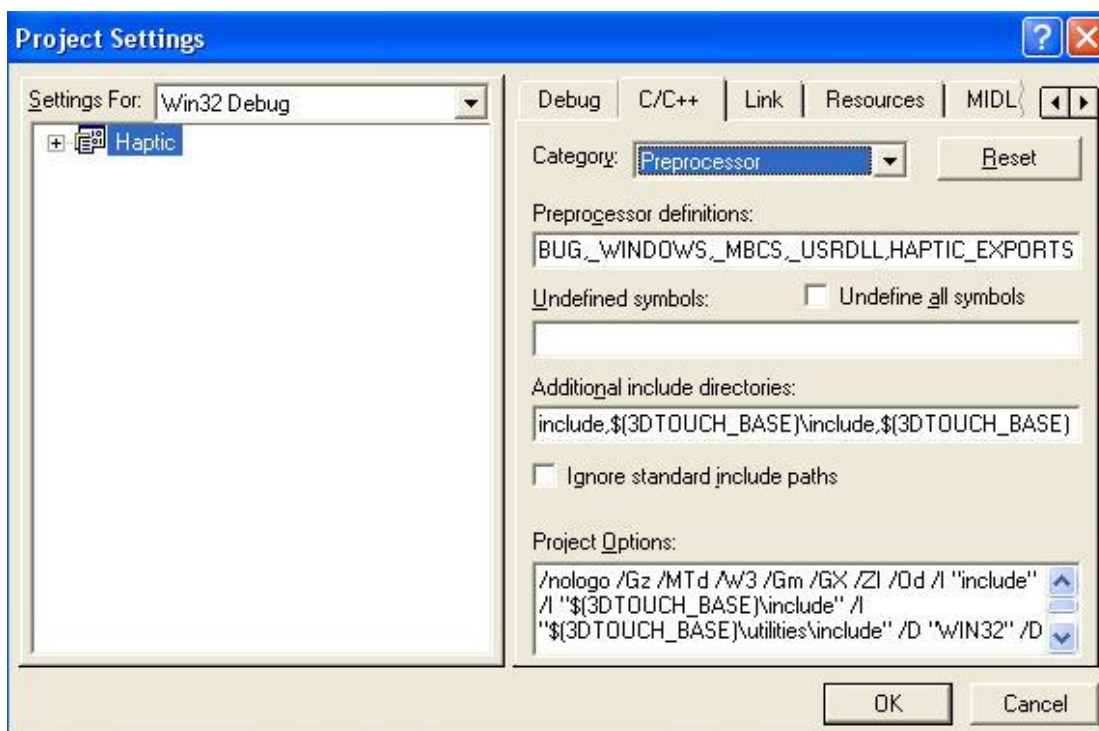


Figura 78 – Janela *Project Settings*, categoria *Preprocessor*

A nomenclatura `3D TOUCH_BASE` é uma variável de ambiente, que atualmente define o seguinte caminho: `C:\Arquivos de programas\SensAble\3D Touch`. Para a próxima etapa da configuração do *Microsoft Visual C++ 6.0*, é preciso acionar a guia *Link*, e na categoria *General*, caixa de texto *Object / library modules*, conforme a Figura 79, inserir as bibliotecas fornecidas pelo fabricante, tais como: *hd.lib*, *hdu.lib*, *snapconstraints.lib*, *glut32.lib*, *hl.lib*, *hlud.lib*, *hdud.lib*. A seleção dessas bibliotecas depende do tipo de aplicação a ser desenvolvida com o dispositivo, sendo que neste exemplo, são inseridas todas elas.

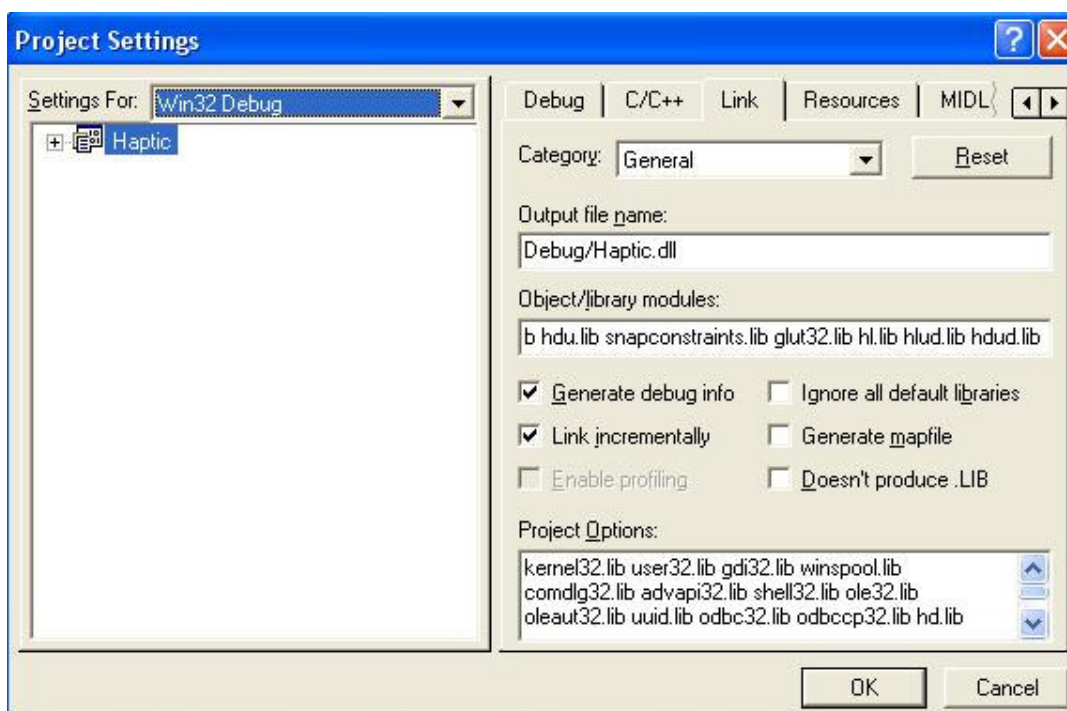


Figura 79 – Inserção de bibliotecas

Para tornar possível a integração, os arquivos *jni.h* e *jni\_md.h*, encontrados no JDK, devem ser inseridos na pasta do *framework*.

No caso da luva de dados, os arquivos *fglove.h* e *fdgloved.dll*, carregados durante a instalação do *software* que acompanha tal dispositivo, devem ser inseridos no diretório anterior ao *framework*, onde encontra-se a aplicação. No compilador, deve ser informada a biblioteca *fgloved.lib* no campo *Object/library modules*, da guia *Link*, categoria *General*, indicando, assim, as funções pré-compiladas da luva de dados, conforme a Figura 80 apresenta.

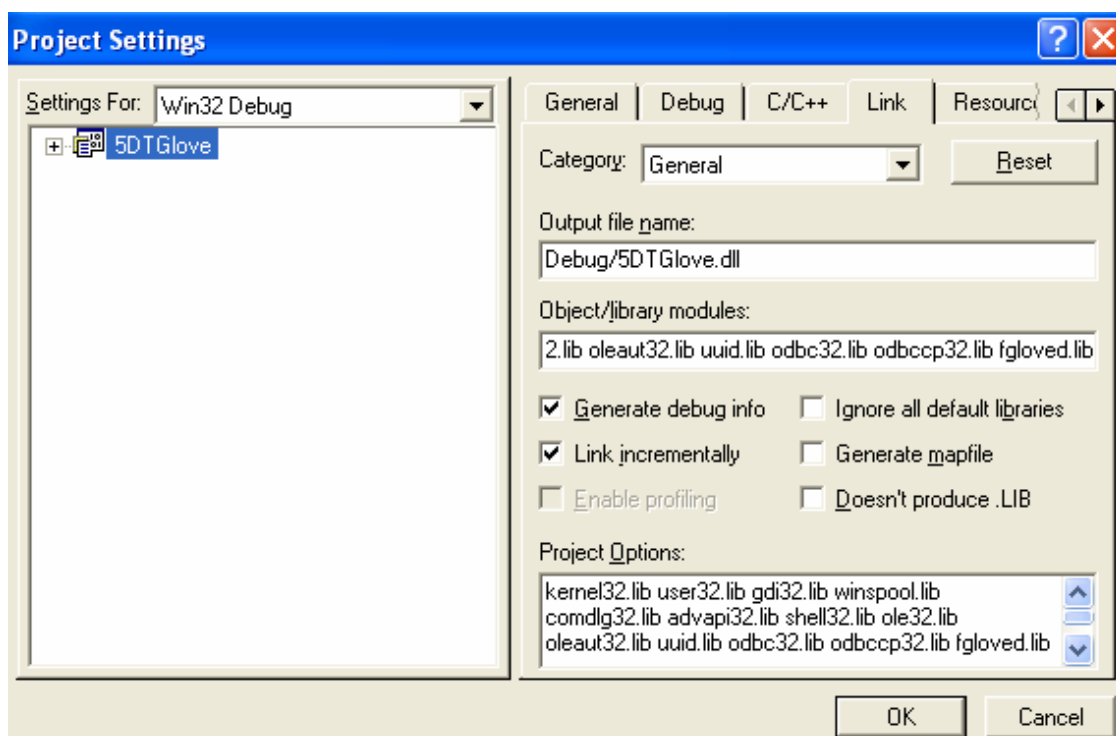


Figura 80 – Configuração do compilador para a luva de dados

O dispositivo háptico, além das configurações descritas para o compilador adotado, necessita de um arquivo de licença de funcionamento, fornecido pelo fabricante, geralmente com o nome de *license.lic*, mediante o preenchimento de um cadastro no *site* deste, onde informações como *HostID* de cada máquina a ser utilizada e o número de série do equipamento são indispensáveis. De posse do arquivo, uma variável de ambiente, chamada de `OH_SDK_LICENSE_PATH`, deve ser configurada indicando sua localização.



## APÊNDICE C

### C.1 *Callback* Principal

Na *callback* principal, o laço de simulação é iniciado com a função *hdBeginFrame* e finalizado com a função *hdEndFrame*. As linhas 3 e 51 do trecho de código definem o início e o fim do laço de simulação respectivamente; na linha 8 é definida a função para obtenção da posição nos três eixos e o armazenamento em *position*; na linha 11 é apresentada a função para obtenção da matriz de transformação e o armazenamento no ponteiro *transform*, que foi definido como um vetor de 16 posições.

Na linha 14 pode-se observar a função do *OpenHaptics Toolkit* para obtenção da velocidade em três direções, e o armazenamento em *velocity*. Esta função foi implementada, entretanto, não foi utilizada no AV, sendo utilizada somente para testes. Nos trechos entre as linhas 17 a 29, e as linhas 32 a 44, são definidas as funções e condições para receber e verificar os estados dos botões 1 e 2, respectivamente. A linha 48 mostra a função para realizar o retorno de força, que é calculado pela parte da aplicação implementada em Java, e transferida para o vetor *force[]*. No trecho entre as linha 54 e 59 está o código para verificar a execução da *callback*, que pode ser finalizada em decorrência de erros.

A função *hdStopScheduler* e *hdDisableDevice*, usadas respectivamente para finalizar o *Scheduler* e desabilitar o dispositivo, são acionadas na execução do método *finalize*, acionado quando o objeto é destruído, e antes que o *Garbage Collector* seja iniciado.

```

0001HDCallbackCode HDCALLBACK MainCallback(void *data)
0002{
0003    hdBeginFrame(hdGetCurrentDevice());
0004
0005    //Funções de estado do OpenHaptic Toolkit
0006
0007    //Função para obter posição
0008    hdGetDoublev(HD_CURRENT_POSITION, position);
0009
0010    //Função para obter matriz de transformação
0011    hdGetDoublev(HD_CURRENT_TRANSFORM, transform);
0012
0013    //Função para obter a velocidade
0014    hdGetDoublev(HD_CURRENT_VELOCITY, velocity);
0015
0016    //Função para obter o estado do botão 1
0017    hdGetIntegerv(HD_CURRENT_BUTTONS, &currentButton1);
0018    hdGetIntegerv(HD_LAST_BUTTONS, &lastButton1);
0019
0020    if ((currentButton1 & HD_DEVICE_BUTTON_1) != 0 &&
0021        (lastButton1 & HD_DEVICE_BUTTON_1) == 0)
0022    {
0023        button1 = 1; //Botão 1 pressionado
0024    }
0025    else if ((currentButton1 & HD_DEVICE_BUTTON_1) == 0 &&
0026            (lastButton1 & HD_DEVICE_BUTTON_1) != 0)
0027    {
0028        button1 = 0; //Botão 1 está liberado
0029    }
0030
0031    //Função para obter o estado do Botão 2
0032    hdGetIntegerv(HD_CURRENT_BUTTONS, &currentButton2);
0033    hdGetIntegerv(HD_LAST_BUTTONS, &lastButton2);
0034
0035    if ((currentButton2 & HD_DEVICE_BUTTON_2) != 0 &&
0036        (lastButton2 & HD_DEVICE_BUTTON_2) == 0)
0037    {
0038        button2 = 1; //Botão 2 está pressionado
0039    }
0040    else if ((currentButton2 & HD_DEVICE_BUTTON_2) == 0 &&
0041            (lastButton2 & HD_DEVICE_BUTTON_2) != 0)
0042    {
0043        button2 = 0; //Botão 2 está liberado
0044    }
0045
0046    //Realiza o retorno de força
0047    //Função para o retorno de força
0048    hdSetDoublev(HD_CURRENT_FORCE, force);
0049
0050
0051    hdEndFrame(hdGetCurrentDevice());
0052
0053    //Verifica erros durante a execução da CALLBACK
0054    if (HD_DEVICE_ERROR(error = hdGetError()))
0055    {
0056        fprintf(stderr, "CALLBACK encerrada");
0057        getch();
0058        return HD_CALLBACK_DONE;
0059    }
0060
0061    return HD_CALLBACK_CONTINUE;
0062}

```

Figura 81 – Código da *Callback* principal

O trecho de código apresentado na Figura 82 é usado para testar o estado de execução da *callback* a cada quadro, podendo causar o encerramento da mesma.

```
0002  if (!hdWaitForCompletion(hMain, HD_WAIT_CHECK_STATUS))
0003  {
0004      verifyThread = -1;
0005  }
0006  else
0007  {
0008      verifyThread = 0;
0009  }
0010
0011  return verifyThread;
```

Figura 82 – Código de verificação da *Callback*

## APÊNDICE D

### D.1 Código para Inicialização da Luva de Dados

```

0001//Função para iniciar comunicação com a luva de dados
0002JNIEXPORT jint JNICALL Java_ViMeT_NativeGlove_openGlove(JNIEnv *env, jobject obj)
0003{
0004    #ifdef WIN32
0005        szPort = "USB";
0006    #else
0007        szPort = "/dev/fglove";
0008    #endif
0009
0010    strcpy(szPortToOpen, szPort);
0011
0012    if (strcmp(szPort, "USB") == 0)
0013    {
0014        unsigned short aPID[5];
0015        int nNumFound = 5;
0016
0017        //Verifica nas portas USB Data Gloves disponíveis
0018        fdScanUSB(aPID, nNumFound);
0019
0020        for (int c = 0; c < nNumFound; c++)
0021        {
0022            printf("Luvas disponíveis nas portas USB:\n");
0023            printf("%i - ", c);
0024            switch (aPID[c])
0025            {
0026                case DG14U_R:
0027                    printf("Data Glove 14 Ultra Right\n");
0028                    break;
0029                case DG14U_L:
0030                    printf("Data Glove 14 Ultra Left\n");
0031                    break;
0032                case DG5U_R:
0033                    printf("Data Glove 5 Ultra Right\n");
0034                    break;
0035                case DG5U_L:
0036                    printf("Data Glove 5 Ultra Left\n");
0037                    break;
0038                default:
0039                    printf("Desconhecida\n");
0040            }
0041        }
0042
0043        sprintf(szPortToOpen, "USB%i", 0);
0044        fdOpen(szPortToOpen);
0045    }
0046
0047    if (NULL == (pGlove = fdOpen( szPortToOpen)))
0048    {
0049        verify = -1;
0050    }
0051    else
0052    {
0053        NumSensors = fdGetNumSensors(pGlove);
0054        verify = 1;
0055    }
0056
0057    return verify;
0058
0059
0060}

```

Figura 83 – Código em C++ para inicialização da luva de dados

Na linha 18 são verificadas as luvas de dados disponíveis nas portas USB; no trecho entre as linhas 20 e 41, é identificado o tipo de luva disponível por meio de comparações com constantes definidas na biblioteca do fabricante (DG14U\_R, DG14U\_L, DG5U\_R,

DG5U\_L), levando em consideração o número de sensores e se é luva para mão direita ou para mão esquerda.

Na linha 48 é realizada a abertura de comunicação com o dispositivo, que é armazenado no ponteiro *pGlove*. Nas linhas seguintes é identificada a ocorrência de problemas na inicialização.

## D.2 Funções Nativas para Obtenção de Informações

O trecho de código para a captação dos valores dos sensores e dos gestos dentro das funções nativas.

```

0001
0002//Obtém o valor de todos os sensores
0003JNIEXPORT jshortArray JNICALL
0004Java_ViMeT_NativeGlove_getRawSensorData(JNIEnv *env, jobject obj)
0005{
0006     jshortArray result = env->NewShortArray(NumSensors);
0007     fdGetSensorRawAll(pGlove,data);
0008     env->SetShortArrayRegion(result,0,NumSensors,(jshort*)data);
0009     return result;
0010}
0011
0012//Obtém o número do gesto pré-definido pelo fabricante
0013JNIEXPORT jint JNICALL
0014Java_ViMeT_NativeGlove_getGesture(JNIEnv *env, jobject obj)
0015{
0016     jint gesture = fdGetGesture(pGlove);
0017     return gesture;
0018}

```

Figura 84 – Código nativo para receber informações da luva de dados

Pode-se observar que a primeira função nativa é um vetor do tipo *short* (linha 3), e o valor a ser retornado em *result*, é um vetor do mesmo tipo, declarado na linha 6 com o tamanho definido de acordo com o número de sensores, informação verificada no início da aplicação. Na linha 7 nota-se a função disponibilizada pelo fabricante para obtenção dos valores dos sensores simultaneamente, armazenando-os no ponteiro *data* (vetor do tipo *short*), e posteriormente transferidos para *result* (linha 8).

Já a segunda função nativa é definida como sendo do tipo *int* e possui a variável do tipo inteiro *gesture*, que recebe o valor da função fornecida pelo fabricante (linha 16) de acordo com o gesto captado, o qual é retornado quando o método nativo é executado no programa escrito em linguagem de programação Java e apresentado no *prompt* de comandos.

## APÊNDICE E

### E.1 Importação e Definição de Características da Mão Virtual

O código apresentado na Figura 85a representa a importação e definição de características dos objetos que vão compor a mão virtual, que é formada pela palma da mão (linhas 2, 3 e 4), o dedo mínimo, com a esfera de rotação (linhas 6, 7 e 8), o corpo do dedo (linhas 10, 11 e 12), e a ponta do mesmo (linhas 14, 15 e 16).

```

0001
0002 objetos[2] = new ObjRig(
0003 "C:\\Documents and Settings\\CLEBER1\\Desktop\\Modelagem 3D\\PalmaDaMao.obj",
0004 Object3D.STEREOCOPY, ObjectFile.RESIZE);
0005
0006 objetos[13] = new ObjRig(
0007 "C:\\Documents and Settings\\CLEBER1\\Desktop\\Modelagem 3D\\EsferaMinimo.obj",
0008 Object3D.STEREOCOPY, ObjectFile.RESIZE);
0009
0010 objetos[7] = new ObjRig(
0011 "C:\\Documents and Settings\\CLEBER1\\Desktop\\Modelagem 3D\\CorpoMinimo2.obj",
0012 Object3D.STEREOCOPY, ObjectFile.RESIZE);
0013
0014 objetos[12] = new ObjRig(
0015 "C:\\Documents and Settings\\CLEBER1\\Desktop\\Modelagem 3D\\PontaMinimo.obj",
0016 Object3D.STEREOCOPY, ObjectFile.RESIZE);

```

Figura 85a – Importação dos objetos

A Figura 85b apresenta a hierarquia de formação de todos os cinco dedos, e a adição dos mesmos ao objeto que representa a palma da mão, bem como, a adição deste último objeto no grafo de cena para apresentação no AV (linha 27). Nas linhas 3 a 5, é possível notar a definição da hierarquia para o dedo mínimo: na linha 3 a esfera de rotação (objetos[13]) é conectada à palma da mão (objetos[2]), na linha 4 o corpo do referido dedo (objetos[7]) é conectado à esfera de rotação (objetos[13]), e na linha 5 a ponta do dedo (objetos[12]) é conectada ao corpo do dedo (objetos[7]).

```

0001
0002     |//Mínimo
0003     objetos[2].getMotionTransform2().addChild(objetos[13].getBranchGroup());
0004     objetos[13].getMotionTransform2().addChild(objetos[7].getBranchGroup());
0005     objetos[7].getMotionTransform2().addChild(objetos[12].getBranchGroup());
0006
0007     //Anelar
0008     objetos[2].getMotionTransform2().addChild(objetos[14].getBranchGroup());
0009     objetos[14].getMotionTransform2().addChild(objetos[6].getBranchGroup());
0010     objetos[6].getMotionTransform2().addChild(objetos[11].getBranchGroup());
0011
0012     //Médio
0013     objetos[2].getMotionTransform2().addChild(objetos[15].getBranchGroup());
0014     objetos[15].getMotionTransform2().addChild(objetos[5].getBranchGroup());
0015     objetos[5].getMotionTransform2().addChild(objetos[10].getBranchGroup());
0016
0017     //Indicador
0018     objetos[2].getMotionTransform2().addChild(objetos[16].getBranchGroup());
0019     objetos[16].getMotionTransform2().addChild(objetos[4].getBranchGroup());
0020     objetos[4].getMotionTransform2().addChild(objetos[9].getBranchGroup());
0021
0022     //Polegar
0023     objetos[2].getMotionTransform2().addChild(objetos[17].getBranchGroup());
0024     objetos[17].getMotionTransform2().addChild(objetos[3].getBranchGroup());
0025     objetos[3].getMotionTransform2().addChild(objetos[8].getBranchGroup());
0026
0027     this.addGrafo(objetos[2].getBranchGroup());

```

Figura 85b – Definição da hierarquia

## E.2 Importação e Definição de Características da Seringa Virtual

A Figura 86 define o código para importação e hierarquização dos objetos na composição da seringa. As linhas 2, 3 e 4 apresentam a importação do objeto virtual que representa o instrumento médico. As linhas 6, 7 e 8 mostram o trecho de código para importação do objeto virtual que representa a esfera de rotação, as linhas 12 e 13 apresentam o trecho para adição dos objetos no AV. A linha 17 indica a hierarquia, onde o instrumento médico (objetos[1]) é conectado à esfera (objetos[2]), que por sua vez, é conectado ao grafo de cena (linha 19), constituindo a cena com o referido objeto.



```
0001
0002 objetos[1] = new ObjRig(
0003 "C:\\Documents and Settings\\CLEBER1\\Desktop\\ObjetosModelados\\seringa pronta.obj",
0004 Object3D.STEREOCOPY, ObjectFile.RESIZE);
0005
0006     objetos[2] = new ObjRig(
0007 "C:\\Documents and Settings\\CLEBER1\\Desktop\\Modelagem 3D\\EsferaSeringa.obj",
0008 Object3D.STEREOCOPY, ObjectFile.RESIZE);
0009
0010
0011     //Adição dos objetos no universo
0012     this.add2(objetos[1]);
0013     this.add2(objetos[2]);
0014
0015
0016     //Seringa
0017     objetos[2].getMotionTransform2().addChild(objetos[1].getBranchGroup());
0018
0019     this.addGrafo(objetos[2].getBranchGroup());
```

Figura 86 – Trecho de código da composição da seringa