

UNIVERSIDADE SAGRADO CORAÇÃO

CARLOS EDUARDO PERRONI TURINI

**REALIDADE VIRTUAL COMO FERRAMENTA DE
APOIO NO DIAGNÓSTICO E PREVENÇÃO DO
CÂNCER DE MAMA**

BAURU

2010

CARLOS EDUARDO PERRONI TURINI

**REALIDADE VIRTUAL COMO FERRAMENTA DE
APOIO NO DIAGNÓSTICO E PREVENÇÃO DO
CÂNCER DE MAMA**

Trabalho de conclusão de curso apresentado ao Centro de Ciências Sociais Aplicadas como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação da Prof.^a Ms. Larissa Pavarini da Luz.

BAURU

2010

CARLOS EDUARDO PERRONI TURINI

**REALIDADE VIRTUAL COMO FERRAMENTA DE APOIO NO
DIAGNÓSTICO E PREVENÇÃO DO CÂNCER DE MAMA**

Trabalho de conclusão de curso apresentado ao Centro de Ciências Sociais Aplicadas como parte dos requisitos para obtenção do título de bacharel em Ciência da Computação, sob orientação da Profª. Ms. Larissa Pavarini da Luz..

Banca examinadora:

Profª. Ms. Larissa Pavarini da Luz
Universidade Sagrado Coração

Prof. Esp. Henrique Pachioni Martins
Universidade Sagrado Coração

Prof. Dr. Elvio Gilberto da Silva
Universidade Sagrado Coração

Bauru, 15 de dezembro de 2010.

Dedico este trabalho aos meus pais,
meu irmão, minha irmã

e minha namorada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, aos meus pais José Eduardo e Joceli por *não terem medidos esforços, por me apoiarem e sempre estarem comigo em todos os momentos. Pelas orações, pelo amor, carinho e confiança e por sempre acreditarem que eu seria capaz de chegar aqui.*

Agradeço à minha namorada Andresa que me apoiou e ajudou na conclusão deste trabalho, dando aquela palavra de apoio nas horas difíceis, e que soube entender minha falta em alguns momentos.

A todos amigos e em especial à Luciana Martins, Jonatas Balderramas, Luciano Pacheco de Almeida Prado e Tais Maran que me acompanharam neste 5 anos de graduação

A minha orientadora neste trabalho a Prof^a. Ms. Larissa Pavarini da Luz que deu o devido suporte para concluir esse trabalho e conseqüentemente o curso.

A todos os professores que passei em especial aos do curso de Ciência da Computação, ao coordenador e professor Elvio Gilberto da Silva que sempre estava lá para resolver os pepinos que apareciam.

Enfim a todos que de uma certa forma colaboraram para a realização desta minha graduação

RESUMO

O câncer de mama tem sido o motivo de milhares de mortes atualmente no mundo. Essa doença tem sua probabilidade de cura aumentada se for diagnosticada precocemente, e por esta razão é proposto neste trabalho o uso de softwares baseados em realidade virtual e programados em Java 3D, na aprendizagem de estudantes dos cursos superiores de medicina. Este sistema pretende dar ao estudante uma visão mais ampla dos órgãos femininos, doenças e distúrbios, podendo vê-los de diversos ângulos e tamanhos.

Palavras Chave: Câncer de mama. Diagnóstico precoce. Estudantes de Medicina. Realidade Virtual. Java 3D.

ABSTRACT

Breast cancer has been the subject of thousands of deaths in the world today. This disease has increased the probability of cure if caught early, and for this reason this work proposes the use of software based on virtual reality and programmed in Java 3D, on learning of students of higher education in medicine. This system will give the student a wider view of the female organs, diseases and disorders, and can see them from different angles and sizes.

Keywords: Breast cancer. early diagnosis. Medical Students. Virtual Reality. Java 3D.

LISTA DE FIGURAS

Figura 1 - Foto promocional de um protótipo do Sensorama.....	24
Figura 2- Possíveis formas de visualização do simulador de anestesia	29
Figura 3 - Interface de treinamento de sutura executado no simulador.....	30
Figura 4 - Simulador de Histeroscopia.....	31
Figura 5 - Ambiente Virtual do Simulador de Histeroscopia.....	31
Figura 6 - (a) Simulador EyeSi , (b) Cenário do EyeSI	32
Figura 7 - (a) Sistema <i>EYESIY</i> , (b) Modelo Mass-Springs	33
Figura 8 - Estrutura de um Grafo de cena	38
Figura 9 - Legenda do Grafo de Cena (SUN, 2010b)	39
Figura 10 - Código para importação de bibliotecas para permitir o carregamento de objetos dentro da API Java 3D	42
Figura 11 - Código responsável pela importação ou carregamento do objeto 3D para a API Java 3D.....	43
Figura 12 - Objeto 3D carregado na API Java 3D	43
Figura 13 - Mamas que serão utilizadas no protótipo	46
Figura 14 - Bibliotecas loaders da API Java 3D.....	47
Figura 15 - Principais bibliotecas utilizadas no protótipo	47
Figura 16 - Trecho inicial do código do protótipo	48
Figura 17 - Trecho do código responsável pela criação do grafo de cena.....	49
Figura 18 - Atribui cor ao background.....	50
Figura 19 0- Cria e atribui cor à luz ambiente	50
Figura 20 - Os três eixos usado para posicionamento dos pontos de luz.....	51
Figura 21 - Criação dos pontos de luzes e posicionamento	51
Figura 22 - URL onde deve ser armazenado o arquivo com extensão .obj.....	52
Figura 23 - Trecho de código onde é carregado o arquivo com extensão .obj.....	53
Figura 24 - Tela inicial do protótipo, com a imagem 3d de uma mama.....	54

LISTA DE TABELAS

Tabela 1 - Tipos de arquivos que podem ser carregados na API Java 3D..... 41

SUMÁRIO

1 - INTRODUÇÃO	11
2 – OBJETIVOS	12
2.1- OBJETIVOS GERAIS	12
2.2 - OBJETIVOS ESPECIFICOS	12
3 – JUSTIFICATIVA	13
4 – ORGANIZAÇÃO DO TRABALHO	14
5 – O CÂNCER DE MAMA.....	15
5.1 – PREVENÇÃO, DIAGNÓSTICO E TRATAMENTO.....	16
6 – REALIDADE VIRTUAL	21
6.1 – HISTÓRICO	24
6.2 – REALIDADE VIRTUAL NA MEDICINA	27
6.2.1 Aplicações Médicas utilizando Realidade Virtual.....	28
7 – IDÉIA DO PROTÓTIPO DE PREVENÇÃO AO CÂNCER DE MAMA.....	33
7.1 – LINGUAGEM DE PROGRAMAÇÃO	34
8 – METODOLOGIA.....	35
9 – TECNOLOGIAS DO PROTÓTIPO	36
9.1 – TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO PROTÓTIPO	36
9.1.1 - Java.....	36
9.1.2 - A API Java 3D.....	37
9.1.3 - Ambiente Virtual	38
9.1.4 - Carregando Modelos Geométricos Prontos	40
9.1.5 – Software de Modelagem Blender	44
10 – DESENVOLVIMENTO DA MODELAGEM E DA PROGRAMAÇÃO JAVA DO PROTÓTIPO.....	45
10.1 – MODELAGEM DO OBJETO VIRTUAL	45

10.2 – PROGRAMAÇÃO DO PROTÓTIPO	46
12 – CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	55
REFERÊNCIAS BIBLIOGRÁFICAS	56
ANEXO	58

1 - INTRODUÇÃO

Devido ao alto índice de mulheres com câncer de mama no mundo, é necessário que exista um plano de prevenção, e que os estudantes que serão futuros médicos possam estar aptos a detectar a doença cada vez mais rápido em seu estágio inicial. Para isso é preciso desenvolver projetos e métodos para agilizar esse processo (INCA 2010).

A área de realidade virtual ultimamente tem se desenvolvido bastante, devido às novas tecnologias descobertas tais como, placas de aceleração, monitores com alta resolução capazes de criar imagens quase perfeitas, sensores e câmeras especiais, seus recursos ficaram ainda mais amplos, assim podendo atingir um leque maior de resultados em diversas áreas, sendo a medicina uma delas. A realidade virtual na medicina tem auxiliado muito em exames, como por exemplo, os exames em 3 dimensões disponíveis atualmente em clínicas e hospitais

Segundo o INCA (2010) o câncer de mama é o segundo tipo mais freqüente no mundo, sendo esta um tipo de doença que se desenvolve rapidamente. Pensando nisso, a utilização de técnicas de realidade virtual propõe ao estudante uma maneira de aprender com maior agilidade e eficiência a diagnosticar com maior precisão a doença.

O protótipo proposto será desenvolvido com a linguagem Java que foi escolhida devido a sua gratuidade e a possibilidade de utilização da API Java 3D, só assim sendo possível a utilização de imagens 3D. O protótipo oferecerá visualizações de diferentes ângulos do órgão feminino, possibilitando o reconhecimento de vários tipos de câncer em diferentes estágios de desenvolvimento.

2 – OBJETIVOS

2.1- OBJETIVOS GERAIS

A construção de um protótipo que auxiliará o estudante no reconhecimento de um câncer através de exemplificações de tumores, podendo o médico, através do mesmo, ensinar seus pacientes uma forma de prevenção e conscientização da doença.

2.2 - OBJETIVOS ESPECIFICOS

Desenvolver um avatar que poderá, com o auxílio do protótipo, demonstrar ao estudante onde e como ocorre o câncer de mama.

Verificar a facilitação na aprendizagem que será decorrência para um melhor preparo profissional aos futuros médicos na área do câncer de mama.

Averiguar o ensinamento que o avatar proporcionará aos pacientes na prevenção do câncer de mama através de técnicas específicas, sendo uma delas, o auto exame.

3 – JUSTIFICATIVA

Devido ao aumento dos casos de câncer de mama, observa-se a importância da realidade virtual com a consequência da construção de um protótipo que permite um reconhecimento imediato, permitindo assim, a prevenção e o diagnóstico específico.

Atualmente existem diversos projetos em desenvolvimento e alguns já estão funcionando, porém devido ao alto custo de implantação destes projetos, eles são pouco aproveitados em universidades que possuem o curso de medicina. Por isso é preciso uma conscientização através de esclarecimentos sobre os programas de realidade virtual que auxiliam tanto no diagnóstico quanto na conscientização de pacientes com câncer, pois é necessário que os pacientes saibam o que realmente têm. Isso só poderá acontecer se houver o processo acima citado, pois assim os médicos estarão mais bem preparados tanto na detecção quanto no ato de explicar a doença aos seus pacientes.

4 – ORGANIZAÇÃO DO TRABALHO

No primeiro capítulo é apresentada a introdução que permite uma visão geral sobre o assunto, assim como, a justificativa pela qual foi escolhido este projeto e os objetivos gerais e específicos que mostra o que o *protótipo* pretende alcançar em seus resultados.

No segundo capítulo é oferecida uma explicação sobre o câncer de mama, citando todos os tipos e dando idéias de prevenção, diagnóstico e tratamento.

Será apresentado no terceiro capítulo toda descrição do que é realidade virtual de maneira geral e na medicina, assim como, será apresentado todo seu histórico e suas aplicações médicas.

A idéia do projeto esta sendo apresentada no quarto capítulo, onde é descrito como será o mesmo e o seu funcionamento.

A modelagem encontra-se no quinto capítulo, onde descreve todo o processo a ser modelado do projeto.

No sexto capítulo encontram-se presente as considerações finais do trabalho.

Por fim, constam as referências bibliográficas no sétimo capítulo.

5 – O CÂNCER DE MAMA

Parece um paradoxo estranho, mas o mesmo órgão que é símbolo de vida e feminilidade é responsável pela morte de meio milhão de mulheres por ano, isso ocorre em decorrência ao câncer de mama. As células epiteliais da mama, algumas vezes se comportam de maneira errada. No acoplamento entre o ducto e o lóbulo, ocorre um distúrbio que tem como consequência o não funcionamento mecanismos de controle local, com o aumento do número de células e desorganização tanto na estrutura como na função.

Paralelamente, é bloqueada ou descontrolada a comunicação intercelular, fazendo com que células que antes não se comunicavam passem a trocar sinais. Ao mesmo tempo, uma síntese e secreção inadequadas das proteases provocam a dissolução da membrana basal, permitindo que as células epiteliais modificadas passem para o estroma. Após a ruptura dos limites da membrana basal e terminada a retroação negativa das células vizinhas, qualquer barreira humana poderá ser invadida pelas células malignas. As maiorias dos cânceres apresentam uma relação direta entre incidência e idade, de forma que se ambas forem lançadas em uma escala logarítmica, a curva ascendente torna-se linear. O câncer de mama se comporta de uma forma atípica (FENTIMAN, 1993).

Ainda segundo Fentiman (1993) as pesquisas sobre o câncer de mama são caracterizadas por esforços prodigiosos e enormes gastos. No que se refere à pesquisa das causas, o trabalho tem sido em grande parte epidemiológico, auxiliado por trabalho da biologia celular e molecular. As aplicações da ciência básica são imprevisíveis. Com a descoberta do papel dos oncogenes e dos fatores e receptores de crescimento nos tumores verdadeiros, derivada do trabalho com retrovírus, estes passaram a ser considerados muito importantes em suas relações com o câncer humano.

De acordo com informações do Instituto Nacional de Câncer (INCA), o câncer de mama é o segundo tipo mais frequente no mundo, sendo o mais comum entre as mulheres, respondendo por 22% dos casos novos a cada ano. Se diagnosticado e tratado oportunamente, o prognóstico é relativamente bom.

No Brasil, as taxas de mortalidade por câncer de mama continuam elevadas, muito provavelmente porque a doença ainda é diagnosticada em estádios avançados. Na população mundial, a sobrevivência média após cinco anos é de 61%.

Relativamente raro antes dos 35 anos, acima desta faixa etária sua incidência cresce rápida e progressivamente. Estatísticas indicam aumento de sua incidência tanto nos países desenvolvidos quanto nos em desenvolvimento. Segundo a Organização Mundial da Saúde (OMS), nas décadas de 60 e 70 registrou-se um aumento de 10 vezes nas taxas de incidência ajustadas por idade nos Registros de Câncer de Base Populacional de diversos continentes.

Podem surgir alterações na pele que recobre a mama. Secreção no mamilo também é um sinal de alerta. O sintoma do câncer palpável é o nódulo (carço) no seio, acompanhado ou não de dor mamária. Podem também surgir nódulos palpáveis na axila (INCA, 2010).

5.1 – PREVENÇÃO, DIAGNÓSTICO E TRATAMENTO.

Segundo o INCA (2010) a estimativa de incidência para o ano de 2010 no Brasil foi de 49.240 casos com um risco estimado de 51 a cada 100 mil mulheres. A alta taxa de incidência, assim como o alto custo do tratamento do domínio em questão, torna oportuno o investimento no diagnóstico precoce e na adoção de medidas preventivas que possam reverter este cenário.

Embora a hereditariedade seja responsável por apenas 10% do total de casos, mulheres com história familiar de câncer de mama, especialmente se uma ou mais parentes de primeiro grau (mãe ou irmãs) foram acometidas antes dos 50 anos, apresentam maior risco de desenvolver a doença.

Esse grupo deve ser acompanhado por médico a partir dos 35 anos. É o profissional de saúde quem vai decidir quais exames a paciente deverá fazer. Primeira menstruação precoce, menopausa tardia (após os 50 anos), primeira gravidez após os 30 anos e não ter tido filhos também constituem fatores de risco para câncer de mama.

Mulheres que se encaixem nesses perfis também devem buscar orientação médica. As formas mais eficazes para a detecção precoce do câncer de mama são os exames clínicos e a mamografia.

O exame Clínico das Mamas (ECM): Quando realizado por um médico ou enfermeira treinada, pode detectar tumor de até 1 (um) centímetro, se superficial. Deve ser feito uma vez por ano pelas mulheres entre 40 e 49 anos.

A mamografia (radiografia da mama) permite a detecção precoce do câncer, ao mostrar lesões em fase inicial, muito pequenas (medindo milímetros). Deve ser realizada a cada dois anos por mulheres entre 50 e 69 anos, ou segundo recomendação médica. É realizada em um aparelho de raios-X apropriado, chamado mamógrafo. A mama é comprimida de forma a fornecer melhores imagens, e, portanto, melhor capacidade de diagnóstico.

Há também outras formas para a detecção do câncer de mama, sendo eles:

- Ultra-som: Utilizado não só nos departamentos de radiologia, como adjuvante da mamografia pelos raios-X, como também na clínica para ajudar na avaliação dos nódulos mamários. É uma boa técnica para diferenciar massas sólidas das líquidas. O ultra-som é melhor para detectar tumores maiores.

- Citologia por aspiração com agulha fina (AAF): após limpar a pele sobre o tumor, é introduzida, com um movimento único e firme, uma agulha de calibre 0,6 – 0,9mm e comprimento padrão, presa a uma seringa de vidro ou de plástico. Ao atingir a massa suspeita, a agulha é passada de 8 a 10 vezes através do tumor.

- Biopsia por punção: Tem por objetivo fazer um diagnóstico histológico de câncer de mama em um paciente ambulatorial com um método suspeito.

- Biopsia por excisão: Existem alguns nódulos mamários, cuja natureza não pode ser determinada, que precisam ser excisados sob anestesia local ou geral. O cirurgião que realiza a biópsia deve considerar qual o tipo de tratamento a ser feito, se o nódulo for um carcinoma.

- Diagnóstico por corte de congelamento: Sucedido pela mastectomia, é um método totalmente inadequado de diagnóstico e tratamento do câncer mamário.

O medo de desenvolver a doença faz com que algumas mulheres recorram a procedimentos radicais e polêmicos. Um deles, a retirada profilática dos seios, é

feito antes de qualquer indício de câncer, baseado nos fatores de risco. O método, no entanto, não é recomendado pelo Instituto Nacional de Câncer (INCA, 2010).

Pesquisas recentes desenvolvidas nos Estados Unidos mostram que esse tipo de intervenção não evita o desenvolvimento da doença. Segundo Sérgio Melo, vice-diretor do Hospital do Câncer III, unidade do INCA especializada no tratamento de câncer de mama, as chances de desenvolver um tumor podem até diminuir, mas o risco continua a melhor estratégia é o diagnóstico precoce.

Chamado de adenomastectomia, o procedimento consiste na substituição das mamas por uma prótese de silicone, preservando os mamilos e grande parte da pele da paciente. Na mastectomia, forma usual de tratamento, todos os tecidos são retirados, o que aumenta a margem de segurança. No entanto, mesmo nesse caso há chances de recidiva.

Segundo o Instituto Nacional do Câncer, outro método radical utilizado para combater o câncer de mama é a retirada dos ovários, que deve ser adotado com critério e, em casos específicos, como nos de mulheres jovens que não respondem às técnicas mais utilizadas. Os ovários são responsáveis pela produção do estrogênio, hormônio responsável pelas características sexuais femininas, que alimenta as células cancerosas. A retirada dos órgãos interrompe o processo de nutrição, contudo, o procedimento serve para tratar, jamais para prevenir o câncer.

Segundo o INCA o auto-exame das mamas não pode ser adotado como estratégia isolada de detecção precoce do câncer de mama. A recomendação é que o exame das mamas pela própria mulher faça parte das ações de educação para a saúde que contemplem o conhecimento do próprio corpo.

As evidências científicas sugerem que o auto-exame das mamas não é eficiente para o rastreamento e não contribui para a redução da mortalidade por câncer de mama. Além disso, o auto-exame traz consigo conseqüências negativas, como aumento do número de biópsias de lesões benignas, falsa sensação de segurança nos exames falsamente negativos e impactos psicológicos negativos nos exames falsamente positivos.

Portanto, o exame das mamas realizado pela própria mulher não substitui o exame físico realizado por profissional de saúde qualificado para essa atividade. Em relação à prevenção, a hormonioterapia é indicada especificamente para mulheres

com altas chances de desenvolver câncer de mama ou que já tiveram a doença em um dos seios. Diferentemente da quimioterapia, que age diretamente nas células cancerosas, a hormonioterapia ataca, além do estrogênio, a progesterona, outro tipo de hormônio que alimenta essas células. O procedimento também combate a aromatase enzima que transforma a gordura em estrogênio.

Segundo o Instituto acima supra citado, o exame clínico das mamas com um acompanhamento médico hoje em dia é a melhor maneira de se defender da doença. O diagnóstico precoce eleva em 90% as chances de cura. Mulheres com 40 anos ou mais devem realizar o exame clínico das mamas periodicamente. Entre 50 e 69 anos, toda mulher deve fazer uma mamografia a cada dois anos, no máximo. Aquelas que tiveram ou têm casos de câncer de mama na família (mãe, irmã, filha etc., diagnosticados antes dos 50 anos), ou que tiveram câncer de ovário ou câncer em uma das mamas, em qualquer idade, devem realizar o exame clínico e mamografia, a partir dos 35 anos de idade, anualmente.

Existem vários tipos de tratamento e são os fatores que definem o que mais adequado em cada caso. Analisa-se o resultado do exame anátomo-patológico da biopsia ou da cirurgia e pede-se exames de laboratório e de imagem para definir qual a extensão do tumor e se houve alojamento em outras partes do corpo.

A cirurgia para o tratamento do câncer de mama pode ser conservadora ou radical. Será conservadora quando retira apenas uma parte da mama (quadrantectomia), e será radical quando retira toda a mama. O tipo de cirurgia varia de caso para caso. No caso da retirada parcial, a cirurgia deverá ser complementada pela radioterapia.

A radioterapia é um tratamento à base de aplicação de radiação direcionada ao tumor ou ao local deste e tem por objetivo, se antes da operação, reduzir o tamanho do tumor, e se após, evitar a volta da doença. A radiação bloqueia o crescimento das células, e deve ser utilizada apenas na área afetada, evitando atingir o tecido normal. O tratamento não apresenta complicações.

Outro tratamento utilizado nos casos de câncer é a quimioterapia. A quimioterapia é o uso de medicamentos extremamente potentes no tratamento do câncer. Também é usado para completar a cirurgia, podendo começar antes ou

após a operação. A quimioterapia age em todo o corpo, visando evitar a volta do tumor e o aparecimento em outros órgãos.

O tratamento normalmente é feito com soro pela via endovenosa. Na maioria das vezes, o tratamento dispensa a internação. Primeiramente, o paciente faz uma consulta médica de rotina e, se estiver tudo normal, recebe o soro durante algumas horas e está liberado para voltar para casa.

Outro procedimento que pode ser útil é a hormonioterapia. Porém, é feito um exame para averiguar a utilidade ou não desse tratamento. O exame consiste na medição na dosagem dos receptores de estrogênios das células do tumor. De acordo com o resultado avalia-se a necessidade ou não da hormonioterapia, que consiste na ingestão de um a dois comprimidos por dia durante não menos que dois anos.

6 – REALIDADE VIRTUAL

Diane Ackerman(1993) afirma, em seu livro *A Natural History of the Senses*, que 70% dos receptores do sentido humano encontram-se nos olhos, tornando-os os grandes “monopolistas dos sentidos” [Jacobson, 1994]. A maioria das informações recebidas pelo ser humano tem a forma de imagens visuais, as quais são interpretadas por um computador extremamente eficiente, o cérebro. Os computadores digitais, por sua vez, interpretam informações fornecidas por algum dispositivo de entrada de dados, como um teclado, por exemplo. Atualmente, a RV permite que computadores e mente humana atuem de forma cada vez mais integrada (Machado, 1995).

O termo Realidade Virtual é creditado a Jaron Lanier, fundador da VPL Research Inc., que no início dos anos 80 apoiou pesquisas sobre o mundo virtual, para diferenciar as simulações tradicionais feitas por computador de simulações envolvendo múltiplos usuários em um ambiente compartilhado [Araújo, 1996]. Pesquisas como a de Myron Krueger, em meados da década de 70, já utilizavam o termo Realidade Artificial, e William Gibson utilizou o termo *cyberspace* em 1984, no seu romance de ficção científica *Neuromancer* (Gibson, 1984; Machover, 1994).

Espaço cibernético (*cyberspace*) foi o termo utilizado para designar uma representação gráfica de dados abstraídos dos bancos de dados de todos os computadores do sistema humano. Gibson descreve que uma rede de computadores universal contendo todo tipo de informações, na qual seria possível “entrar” e explorar os dados de forma multisensorial, e onde pessoas com implantes em seus corpos podiam transmitir informações diretamente para o computador. Na verdade, o Espaço Cibernético é um espaço imaginário, uma simulação 4D do espaço-tempo controlada pela interface de realidade virtual [Adams, 1994]. Vince [1995] afirma que, desde que os sistemas de realidade virtual criem o espaço cibernético, é possível interagir com tudo e com todos em um nível virtual.

O termo realidade virtual é bastante abrangente, e acadêmicos, desenvolvedores de *software* e pesquisadores tendem a defini-lo com base em suas próprias experiências, gerando diversas definições na literatura. Pode-se dizer, de uma maneira simplificada, que RV é a forma mais avançada de interface do usuário

com o computador até agora disponível [Hancock, 1995]. Trata-se de uma *interface* que simula um ambiente real e permite aos participantes interagirem com o mesmo [Latta, 1994], permitindo às pessoas visualizarem, manipularem e interagirem com representações extremamente complexas [Aukstakalnis, 1992]. Ela é um paradigma pelo qual usa-se um computador para interagir com algo que não é real, mas que pode ser considerado real enquanto está sendo usado [Hand, 1994].

Outra definição é o uso de computadores e *interfaces* com o usuário para criar o efeito de mundos tridimensionais que incluem objetos interativos com uma forte sensação de presença tridimensional [Bryson, 1996]. Além disso, a realidade virtual engloba um conjunto de técnicas e ferramentas gráficas 3D que permite aos usuários interagir com um ambiente gerado por computador, em tempo real, com uma pequena ou nenhuma consciência de que está usando uma interface usuário-computador [Leston, 1996].

Outros autores [Burdea, 1994; Jacobson, 1991; Krueger, 1991] afirmam que realidade virtual é uma técnica avançada de interface que permite ao usuário realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais. Ainda outra definição é como a simulação do espaço x tempo 4D, isto é, uma animação de pontos de observação apresentada em um contexto interativo e em tempo real. É uma *interface* que proporciona controles para o usuário manipular e interagir com uma base de dados que é o espaço tempo 4D, incluindo a realidade artificial (espaço virtual) e as entidades (objetos virtuais) que ela contém [Adams, 1994]. O termo espaço-tempo 4D geralmente refere-se a imagens computadorizadas 3D animadas às quais foi acrescentada a quarta dimensão, que é o tempo.

Na prática, a realidade virtual permite que o usuário navegue e observe um mundo tridimensional, em tempo real e com seis graus de liberdade. Isso exige a capacidade do *software* de definir, e a capacidade do *hardware* de reconhecer, seis tipos de movimento: para frente/para trás, acima/abaixo, esquerda/ direita, inclinação para cima/ para baixo, angulação à esquerda/à direita e rotação à esquerda/à direita Von Schweber (1995). Na essência, a realidade virtual é um “espelho” da realidade física, na qual o indivíduo existe em três dimensões, tem a sensação do tempo real e a capacidade de interagir com o mundo ao seu redor. Os equipamentos de RV

simulam essas condições, chegando ao ponto em que o usuário pode “tocar” os objetos de um mundo virtual e fazer com que eles respondam, ou mudem, de acordo com suas ações.

Segundo Kirner(1996), a *interface* em RV envolve um controle tridimensional(3D) altamente interativo de processos computacionais. O usuário entra no espaço virtual das aplicações e visualiza, manipula e explora os dados da aplicação em tempo real, usando seus sentidos, particularmente os movimentos naturais tridimensionais do corpo. A grande vantagem é que o conhecimento intuitivo do usuário sobre o mundo físico pode ser transportado para o mundo virtual. Para suportar esse tipo de interação o usuário utiliza dispositivos não convencionais, como capacetes de visualização e controle, e luvas de dados, chamadas *datagloves*. O uso desses dispositivos dá ao usuário a impressão de que a aplicação está funcionando no ambiente tridimensional real, permitindo a exploração do ambiente e a manipulação natural dos objetos com o uso das mãos [Kirner, 1996].

Leston, (1996) diz que o termo Mundo Virtual é usado para denotar o mundo digital criado a partir de técnicas de Computação Gráfica. Vince(1995)Uma vez que é possível interagir e explorar esse mundo por meio de dispositivos de entrada e de saída, ele se transforma em um ambiente virtual, ou ambiente de Realidade Virtual. A RV é, frequentemente, confundida com animação, CAD (*Computer Aided Design*) ou multimídia. Em comparação com essas tecnologias, a RV é orientada ao usuário, ou seja o observador da cena virtual que poderá interagir com o mesmo, a RV é mais imersiva e interativa, porque o usuário que está interagindo com o ambiente virtual se sente dentro dele e pode mudar e influenciar no comportamento dos objetos virtuais e por ultimo mais intuitiva, pois quase não existe dificuldade de interação entre a máquina e o usuário.

Outro fator que podemos destacar é o fato de realidade virtual possuir *rendering*, que é um processo de transformação dos modelos em imagens, feito em tempo real, possibilita que as imagens sejam atualizadas assim que a cena sofre qualquer tipo de modificação, e ainda possui uma descrição funcional dos objetos.

Segundo Kirner(1996) e Bishop(1992), o desenvolvimento de um sistema de realidade virtual necessita de estudos e recursos ligados a percepção sensorial, *hardware*, *software*, interface com o usuário, fatores humanos e aplicações. É

necessário, também, algum domínio sobre dispositivos não convencionais de Entrada/Saída, computadores de alto desempenho, sistemas paralelos e distribuídos, modelagem geométrica tridimensional, simulação em tempo real, navegação, detecção de colisão, avaliação, impacto social e projeto de *interfaces*.

6.1 – HISTÓRICO

A RV começou na indústria de simulação, com os simuladores de vôo que a força aérea do Estados Unidos passou a construir logo após a Segunda Guerra Mundial [Jacobson, 1994]. A indústria de entretenimento também teve um papel importante, ao construir um simulador chamado Sensorama (Figura 1). O Sensorama era uma espécie de cabine que combinava filmes 3D, som estéreo, vibrações mecânicas, aromas, e ar movimentado por ventiladores; tudo isso para proporcionar ao espectador uma viagem multisensorial [Pimentel, 1995]. Patentado em 1962 por Morton Heilig, o equipamento já utilizava um dispositivo para visão estereoscópica.



**Figura 1 - Foto promocional de um protótipo do Sensorama.
(NETTO; MACHADO; OLIVEIRA, 2002)**

Os primeiros trabalhos científicos na área surgiram em 1958, quando a Philco desenvolveu um par de câmeras remotas e o protótipo de um capacete com monitores que permitiam ao usuário um sentimento de presença quando dentro de um ambiente [Comeau, 1961]. Posteriormente, esse equipamento passou a se chamar *head-mounted display*, ou simplesmente HMD [Ellis, 1994].

Alguns anos depois, por volta de 1965, Ivan Sutherland, conhecido como o precursor da RV [Hand, 1994], apresentou à comunidade científica a idéia de desenhar objetos diretamente na tela do computador por meio de uma caneta ótica, marcando o início da Computação Gráfica. Sutherland tornou-se o precursor da atual indústria de CAD e desenvolveu o primeiro vídeo-capacete totalmente funcional para gráficos de computador no projeto “The Ultimate Display”. Esse vídeo-capacete permitia ao usuário observar, movimentando a cabeça, os diferentes lados de um cuborepresentado em uma estrutura fio-de-aramé flutuando no espaço [Fisher, 1990; Machover, 1994].

Na mesma época em que Sutherland criava seu vídeo-capacete na Universidade de Utah, Myron Krueger experimentava combinar computadores e sistemas de vídeo, criando Realidade Artificial na Universidade de Wisconsin [Pimentel, 1995]. Em 1975, Krueger criou o VIDEOPLACE, no qual uma câmera de vídeo capturava a imagem dos participantes e projetava-a em uma grande tela. Os participantes podiam interagir uns com os outros e com objetos projetados nessa tela, sendo que seus movimentos eram constantemente capturados e processados. Essa técnica tornou-se conhecida como Realidade Virtual de Projeção [Jacobson, 1994].

Em 1982, Thomas Furness demonstrava para a Força Aérea Americana o VCASS (*Visually Coupled Airborne Systems Simulator*), conhecido como “Super Cockpit”. Trata-se de um simulador que usava computadores e vídeocapacetes interligados para representar o espaço 3D da cabine de um avião. Os vídeos-capacete integravam as componentes de áudio e vídeo. Assim, os pilotos podiam aprender a voar e lutar em trajetórias com 6 graus de liberdade (6DOF) sem decolar verdadeiramente. O VCASS possuía alta qualidade de resolução nas imagens e era bastante rápido na atualização de imagens complexas. No entanto, o custo

representava um problema: milhões de dólares eram necessários apenas para o capacete [Pimentel, 1995].

Com a nova tecnologia de visores de cristal líquido (LCD), Michael McGreevy começou a trabalhar no projeto VIVED (*Virtual Visual Environment Display*) em 1984 na NASA, no qual seriam geradas imagens estereoscópicas. A resolução das imagens era limitada em comparação ao VCASS, mas o custo era bastante atrativo [Rheingold, 1991]. As componentes de áudio e vídeo foram montadas sobre uma máscara de mergulho utilizando dois visores de cristal líquido com pequenos auto-falantes acoplados. Scott Fisher juntou-se a esse projeto em 1985, com o objetivo de incluir nele luvas de dados, reconhecimento de voz, síntese de som 3D, além de dispositivos de *feedback* (resposta) tátil.

Em 1985, Thomas Zimmerman e Jaron Lanier fundam a VPL Research, tendo como primeiro produto uma luva de dados, chamada *DataGlove*, desenvolvida por Zimmerman e capaz de captar a movimentação e inclinação dos dedos da mão. No mesmo ano uma dessas luvas foi comprada para o projeto VIVED.

No final de 1986 a equipe da NASA já possuía um ambiente virtual que permitia aos usuários ordenar comandos pela voz, escutar fala sintetizada e som 3D, e manipular objetos virtuais diretamente por meio do movimento das mãos. O mais importante é que esse trabalho permitiu verificar a possibilidade de comercialização de um conjunto de novas tecnologias, tornando mais acessível o preço de aquisição e desenvolvimento. A conscientização de que os empreendimentos da NASA poderiam gerar equipamentos comercializáveis deu início a inúmeros programas de pesquisa em RV no mundo inteiro. Desde firmas de *software* até grandes corporações de informática começaram a desenvolver e a vender produtos e serviços voltados para RV. Em 1989 a *AutoDesk* apresentou o primeiro sistema de RV para computadores pessoais (PC) [Jacobson, 1994].

6.2 – REALIDADE VIRTUAL NA MEDICINA

A realidade virtual tem ajudado ultimamente na construção de diversas aplicações que auxiliam no aprendizado de estudantes de medicina, isso acontece porque a RV cria mundos virtuais muito parecidos quando não idênticos ao mundo real, por essa razão tais simulações em RV podem ser usadas no treinamento de médicos.

Machado (2003) afirma que no ensino e treinamento de futuros profissionais da área médica o uso de ferramentas baseadas em RV pode criar uma nova forma de aprendizagem, pois irá oferecer imagens em 3 dimensões, podendo o estudante explorar o mundo virtual em que se encontra com sensações táteis e motoras combinadas, tais recurso podem oferecer um mundo virtual mais real.

Segundo Montero (2003) as aplicações em realidade virtual são muitas e a maioria dos profissionais concordam em dizer que ajuda no processo de ensino e aprendizagem, tendo sua eficácia comprovada. Ainda segundo o autor, que, o nível de eficácia que a aplicação irá oferecer depende muito, da tecnologia que foi empregada, no desenvolvimento, no dinheiro que foi investido e vários outros fatores.

Segundo o autor supracitado em países desenvolvidos a realidade virtual tem sido muito empregada na educação médica, como por exemplo na simulação de operações e cirurgias vídeo-láparoscópica, ensino de anatomia, tudo isso com a vantagem de poder treinar inúmeras vezes sem limite de tempo e utilização, todos os órgãos podem ser vistos tridimensionais e com semi-transparência, a ligação entre eles também podem ser vistas, sendo que pode-se selecionar visões seletivas do corpo e observar vários órgãos ao mesmo tempo. Tais aplicações podem resolver muitos problemas na aprendizagem e na vida de um médico, substituindo métodos que estão falhando tanto na aprendizagem médica quanto profissionalmente, porém devido a situação financeira de alguns países e ao alto custo destas tecnologias, o risco de utilizarem soluções com uma relação custo/benefício menor é muito alta.

Ainda segundo o autor a cima, no Brasil existem instituições que desenvolvem projetos de realidade virtual para o ensino e treinamento na área médica, mas são

poucas, entretanto em alguns governos estaduais estão surgindo propostas para financiar tais projetos, e ai sim eliminar a maior parte dos obstáculos encontrados pelos desenvolvedores destes projetos, que são mão de obra qualificada e softwares apropriados.

6.2.1 Aplicações Médicas utilizando Realidade Virtual

Segundo Liu (2003) os simuladores de cirurgia feitos com principios de realidade virtual fornecem para o médico um maneira de treinar os procedimentos em pacientes virtuais inúmeras vezes, essas simulações auxilia os futuros médicos a assimilar e memorizar idéias de formatos dos órgãos, suas ligações e funções, já o treinamento médico para tal procedimento necessita de simuladores de cirurgia. A realidade virtual também pode ser usada para desenvolvimento de ambientes virtuais(AMBIENTE VIRTUAL) para auxilio no tratamento de fobias. Machado et al.(2004) definiram as aplicações de realidade virtual na área médica em três grupos: planejamento, assistência e treinamento.

Os sistemas de planejamento permitem o estudo de um caso específico e geralmente utilizam imagens de ressonância magnética ou tomografia computadorizada do paciente para gerar uma réplica virtual da situação real.Os sistemas para assistência, por sua vez, são utilizados para dar suporte a um procedimento real, adicionando e sobrepondo elementos virtuais a uma situação real. Por fim, os sistemas de treinamento objetivam a incorporação de habilidades específicas e utilizam ambientes virtuais visando preparar o usuário para realizar um determinado procedimento, podendo simular situações genéricas (desassociadas das peculiaridades de um paciente específico) com alto grau de realismo. (MACHADO *et al.*, 2004)

Continuando Netto *et al.*(2002) citam que o grupo de sistemas de treinamento exerce a função somente em habilidades específicas, utilizando ambientes virtuais visam preparar melhor o usuário e futuro médico para determinado procedimento, podendo ele estar preparado para diversas situações, pois o alto grau de realismo garante isso. Segundo Machado et al. (2004), a parte mais difícil de um sistema de realidade virtual na área médica é aproximar o mundo virtual do mundo real, assim reproduzindo exatamente o que um profissional da medicina sente e vê na vida real.

Com a finalidade de aumentar a qualidade no treinamento do procedimento de anestesia regional foi desenvolvida uma aplicação de RV destinada aos médicos residentes. A aplicação enfoca o treinamento do procedimento em pacientes com câncer abdominal, que não podem fazer cirurgia ou não toleram o uso de drogas, estas já não surtem mais efeito. O simulador de anestesia regional tem como objetivo criar um ambiente virtual realístico para que médicos residentes possam treinar o procedimento em um paciente virtual.

Segundo Blezek *et al.* (2000), o sistema possui duas funções primárias: visualização e *feedback* de força. Para a imersão do usuário no ambiente virtual é usado um HMD (*Head Mounted Display*) e o dispositivo de *force feedback* PHANTON. Para que a exibição seja rápida e a manipulação de estruturas anatômicas realística é utilizado o VHM (*Visible Human Male*), que consiste em um banco de dados de imagens tridimensionais, construídas a partir de imagens médicas provenientes de cadáveres (NLM, 1990). Na Figura 2 ilustra as possíveis formas de visualização do Ambiente Virtual.

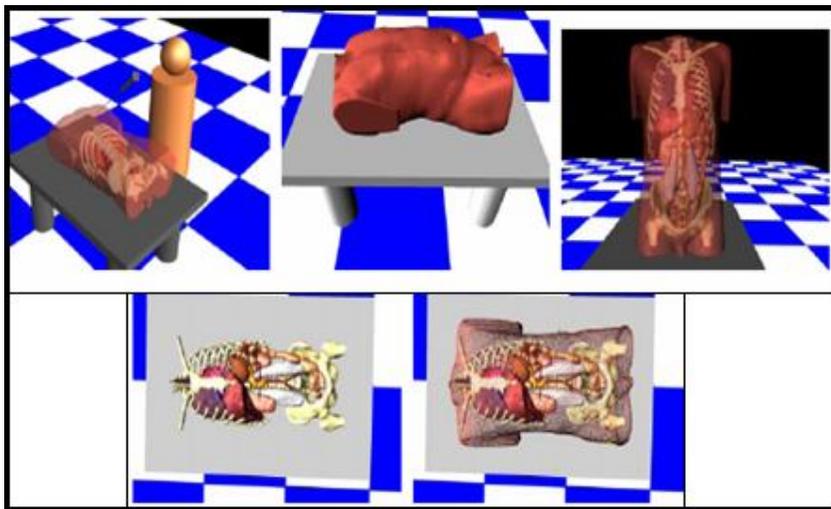


Figura 2- Possíveis formas de visualização do simulador de anestesia (BLEZEK et al., 2000)

Uma outra aplicação de realidade virtual voltada para o treinamento médico é o simulador háptico de sutura. Segundo Webster *et al.* (2001), a intenção era desenvolver um simulador realístico, simples de operar, econômico e com a possibilidade de difundir o seu uso, necessitando apenas de um computador pessoal para utilizá-lo. O simulador possui componentes que modelam e deformam a pele, o tecido e o material de sutura em tempo real, além de registrar todos os estados de atividade durante a tarefa.

O *software* de simulação calcula forças de contato e gera deslocamentos de tecido. Os cálculos da força de resistência variam de acordo com a profundidade e o ângulo de inserção da agulha. A força também muda quando a agulha perfura a pele virtual, quando sai da pele e quando é necessário apertar o material da sutura. Na Figura 3 na página seguinte mostra-se a *interface* do simulador.



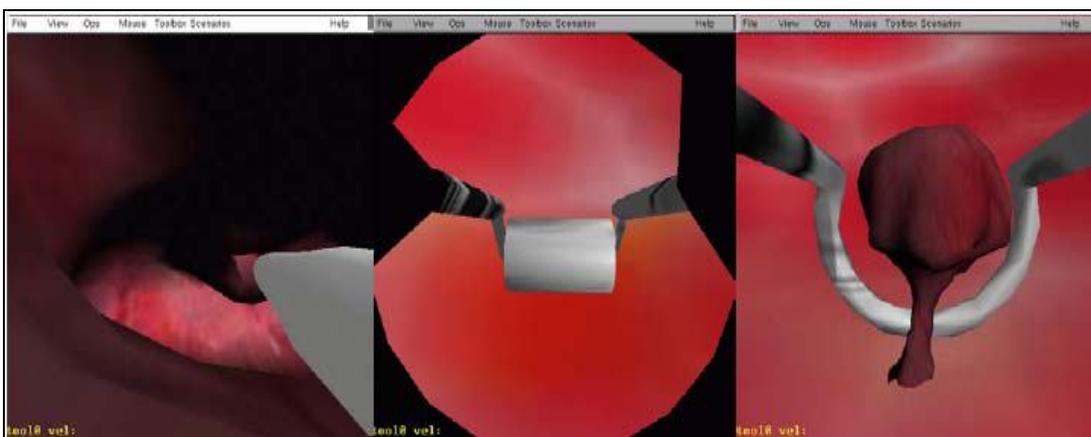
Figura 3 - Interface de treinamento de sutura executado no simulador (WEBSTER et al., 2001)

Montgomery *et al.* (2001) apresentam um simulador cirúrgico baseado em realidade virtual para histeroscopia diagnóstica e cirúrgica. Tal simulador pode ser usado para diversos procedimentos como, análise e planejamento cirúrgico, treinamento de micro cirurgia e simulação de suturas. Segundo Afonso (2000), histeroscopia trata-se de um exame de três fases: passagem do canal cervical, inventário da cavidade uterina (vista panorâmica e com detalhes no aumento de

mais ou menos 20 vezes) e revisão cervical. O sistema descrito permite ao usuário visualizar dados estereoscópicos ou monoscópicos, como objetos *wireframe* (visualização dos pontos e linhas que descrevem o modelo). Possui também uma integração entre o modelo anatômico 3D e os dispositivos hápticos especializados (Figura 4). O simulador ainda possui uma *interface* que permite ao usuário a seleção de vários procedimentos, a apresentação de dados relativos ao exame, além de cronometrar a duração do procedimento e informar a força utilizada durante as ações executadas.



**Figura 4 - Simulador de Histeroscopia
(Adaptado de MONTGOMERY et al., 2001)**



**Figura 5 - Ambiente Virtual do Simulador de Histeroscopia
(MONTGOMERY et al. 2001)**

Esse simulador oferece ao usuário simulações realísticas de procedimento cirúrgicos usados durante alargamento cervical e da histeroscopia, como a remoção de lesões intra-uterinas e a separação e ressecção endometrial. Esta simulação permite a transferência das habilidades virtuais para a melhoria dos procedimentos reais. Na Figura 5 na página seguinte mostra-se o ambiente virtual do simulador em três momentos: o momento de alargamento cervical, da seleção do endométrio e da ressecção.

Pode ser usado como exemplo um simulador chamado EyeSi: trata-se de um sistema que tem o objetivo de simular cirurgias intra-oculares. Segundo Wagner *et al.* (2002), o projeto é baseado em uma instalação mecânica completamente modelada para fornecer ao cirurgião em treinamento todas as percepções sensoriais do olho, como pode ser observado na Figura 6. O olho mecânico tem o mesmo grau de liberdade de rotação do olho humano, sendo que o efeito dos músculos é modelado por um conjunto de molas sobre cada eixo de rotação.

Para a deformação foi utilizado um método híbrido que inclui o método de elementos finitos e o método *Mass Spring* (Massa-Mola). Para a detecção de colisão foi utilizada parte do pacote de detecção de colisão *I-Collide* (CHUNG e WANG, 1996 *apud* WAGNER, 2002), junto com o método *boundingbox*. A visualização realista do cenário de operação inclui efeitos de iluminação e sombras como pode ser visto na Figura 6 (b).

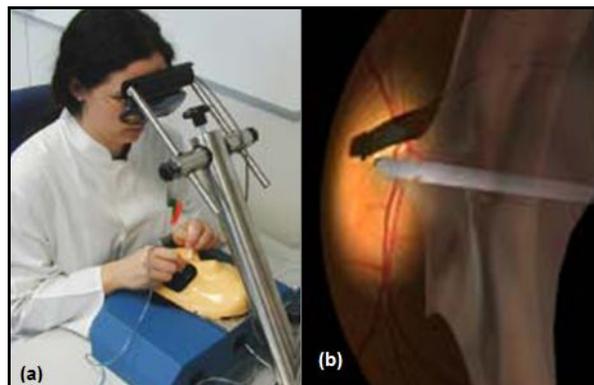


Figura 6 - (a) Simulador EyeSi , (b) Cenário do EyeSI
(WAGNER *et al.*, 2002)

Webster *et al.* (2005) desenvolveu uma aplicação específica para simular o procedimento de Extração Extra-Capsular do Cristalino (EECC). Esta aplicação utiliza o sistema *EYESI/VTM*, conforme mostrado na Figura 7(a). Na Figura 7(b) na página seguinte, mostra-se o modelo utilizado para deformação. Esta aplicação possui praticamente as mesmas características da aplicação construída por Wagner *et al.* (2002).

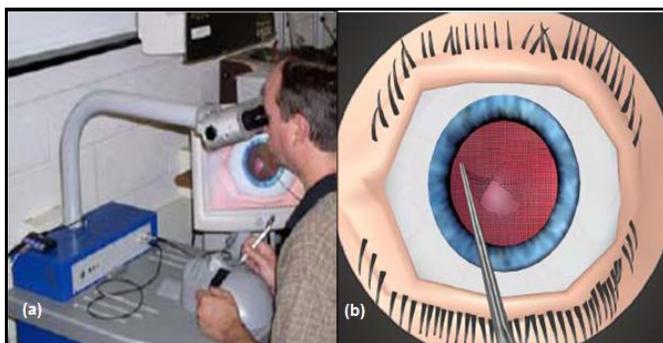


Figura 7 - (a) Sistema *EYESI/VTM*, (b) Modelo Mass-Springs- usado para romper a membrana (WEBSTER *et al.*, 2005)

Um exemplo utilização de RV na Medicina são as aplicações para reabilitação.

Niniss e Inoue (2006) desenvolveram um sistema de RV para auxiliar deficientes físicos que utilizam cadeiras de rodas. Nesse sistema é criado um Ambiente virtual que simula um local que o paciente já conhece. Por meio de dispositivos específicos o paciente pode explorar o Ambiente Virtual para reconhecimento, familiarização com a cadeira de rodas e, principalmente, a reabilitação motora.

7 – IDÉIA DO PROTÓTIPO DE PREVENÇÃO AO CÂNCER DE MAMA

O projeto que será desenvolvido auxiliará estudantes do curso de medicina a treinarem como executar o diagnóstico e a prevenção do câncer de mama, através deste software o estudante poderá interagir com o avatar podendo visualizar tipos de câncer e distúrbios de diferentes ângulos e tamanhos, além disso, o software

disponibilizará alternativas de prevenção, como por exemplo a execução do auto exame (como se faz).

7.1 – LINGUAGEM DE PROGRAMAÇÃO

A linguagem utilizada no *protótipo* proposto será a Java 3D API. Ela é um pacote disponível em Java. Utilizando construtores Java 3D API, será possível criar um mundo virtual 3D no qual pode construir e manipular estruturas 3D. Semelhantemente com a Java 3D API, também armazenadas as informações de cena 3D que são visualizada em uma aplicação, como objetos em espaço em uma hierarquia de nós conhecida como um gráfico de cena. Os nós representam objetos, informação sobre posição ou movimento, e informação sobre aparência e iluminação. Uma visão mais detalhada da linguagem usada no protótipo será exposta mais a diante nos próximos capítulos.

8 – METODOLOGIA

O trabalho em questão tem por objetivo mostrar a importância da Realidade Virtual no diagnóstico e prevenção do câncer de mama.

Busca-se em primeiro plano, o conhecimento através de uma fundamentação teórica advindo do câncer de mama, apresentando-nos a prevenção, diagnóstico e tratamentos. Por essas razões, o meio pelo qual auxiliará na construção de tais conhecimentos, será através de artigos científicos e dados coletados por órgãos públicos responsáveis e também através de artigos médicos divulgados em jornais, revistas e online. Tais dados coletados servirão como base inicial para a realização dessa pesquisa.

Em seguida, será implementada a parte prática, ou seja, será desenvolvido um software embasado nos problemas iniciais existentes no aprendizado de estudantes do curso superior de medicina, sendo eles, a prevenção e o diagnóstico do câncer de mama, onde proporcionará uma melhor visualização do órgão em questão para um melhor esclarecimento frente a importância da Realidade Virtual na medicina.

O objeto 3D será desenvolvido no *software* de modelagem Blender, que é um *software* de distribuição gratuita e que não necessita a aquisição de uma licença, disponibiliza uma gama de recursos de modelagem em 3D, este aplicativo funciona na maioria dos sistemas operacionais existentes. Nele será feita toda a parte de modelagem do avatar e ambiente virtual.

Toda a programação do ambiente virtual e do objeto 3D será realizada em Java, utilizando como auxílio a API Java 3D e de suas classes pré definidas, como por exemplo, a classe *loader*, também chamada de carregador, ela será utilizada no carregamento do objeto 3D, ou seja, o Objeto 3D após ter sido modelado e exportado pelo blender como arquivo wavefront com extensão *.obj* será carregado pelo Java utilizando a classe *loader*.

9 – TECNOLOGIAS DO PROTÓTIPO

Neste capítulo será descrita todas as tecnologias utilizadas para o desenvolvimento do protótipo proposto nos capítulos anteriores.

9.1 – TECNOLOGIAS UTILIZADAS PARA O DESENVOLVIMENTO DO PROTÓTIPO

Para a implementação do protótipo foram utilizadas a linguagem de programação Java (versão JDK 1.6.0_21) (SUN, 2010a) e a API Java3D (versão 1.5.1) (SUN, 2010b). Suas principais características são descritas nas seções seguintes.

9.1.1 - Java

A linguagem Java e o conjunto de bibliotecas API Java3D (será explicada nas próximas seções) foram escolhidas para serem utilizadas neste protótipo devido ao fato da gratuidade, podendo ser utilizadas em vários locais sem a necessidade de se comprar uma licença. Outra razão é que a API Java3D possui várias classes que auxiliam na implementação do Ambiente Virtual, interação e visualização tridimensional.

Segundo Deitel e Deitel (2010), a linguagem Java possui muitas vantagens uma delas é toda a segurança semelhante a uma linguagem compilada. A forma de se criar ponteiros é simplificada, possui também coleta automática de lixo, é multitarefa e permite o controle de *threads*¹. Oferece grande suporte para aplicações de pequeno, médio e grande porte para Internet e Intranet. Ainda segundo os autores supra citados, citam como desvantagens da linguagem: lentidão comparada com outras linguagens pelo fato de ser interpretada e não compilada.

¹Threads: o objeto do Java proveniente da classe `java.lang.Thread`, representa uma instância da CPU da máquina virtual Java, e que tem associada um trecho de código que será executado, e uma área de memória.

Portanto mesmo com algumas características negativas a linguagem Java possui diversas vantagens para desenvolvimento de aplicações, como a portabilidade para outras plataformas e sistemas operacionais (Windows, Linux, Unix e MacOS), estabilidade e um conjunto de bibliotecas com recursos de comunicação e gráficos, como, por exemplo, a API Java 3D, que foi utilizada no desenvolvimento do protótipo. A documentação de Java, fornecida pela SUN (2010), afirma que devido às características de orientação a objetos, sua interface contribui para reutilização de código, utilização de bibliotecas de terceiros com proteção e encapsulamento e, além disso, oferece robustez e segurança. Mais aplicabilidades do Java serão abordadas na próxima seção.

9.1.2 - A API Java 3D

Segundo a Sun(2010) a *API Java 3D que significa Application Programming Interface*, ou seja traduzindo para o português seria Interface de Programação de Aplicativos e foi desenvolvida pela *Sun Microsystems*, para renderizar gráficos interativos 3D usando a linguagem de programação Java. A *API Java 3D* oferece um conjunto de classes que permite o desenvolvimento de aplicações 3D em alto nível, utilizando-se de recursos como criação e manipulação de geometrias 3D, animações e, ainda, interatividade com dispositivos convencionais e não convencionais. São definidas mais de 100 classes apresentadas no pacote *javax.media.j3d.*, possibilitando que o desenvolvimento de aplicações 3D seja totalmente orientado a objetos.

A *API Java3D* auxilia através de suas classes pré-definidas pela SUN a criar ambientes Virtuais usando o conceito de grafo de cena, uma estrutura hierárquica organizadora dos objetos que compõem e definem uma cena, preservando as relações espaciais existentes entre eles. Tais objetos referem-se tanto às geometrias e suas propriedades, como às informações necessárias para que a cena seja renderizada de um ponto particular de visão (SUN, 2010).

O grafo de cena é composto por objetos das classes *VirtualUniverse*, *LocaleSceneGraphObject*, *Node*, *NodeComponent*, *Group*, *Leaf* , *BranchGroup*, *TransfomGroup*, *Shape3D*, *Light*, *Behavior*, *Sound*, *Geometry*, *Appearance*, *Material*,

Texture entre outras.No capítulo seguinte será abordado de maneira detalhada as características de um grafo de cena.

9.1.3 - Ambiente Virtual

Um ambiente virtual é um Universo Virtual, e segundo Mansour(2003) é nele que estão contidos todos os objetos 3D juntamente com as luzes de qualquer ambiente 3D. O universo é composto por um ou mais grafos de cena, que organizam os objetos em uma estrutura tipo árvore. A estrutura de um grafo de cena esta descrita de maneira mais detalhada na Figura 8.

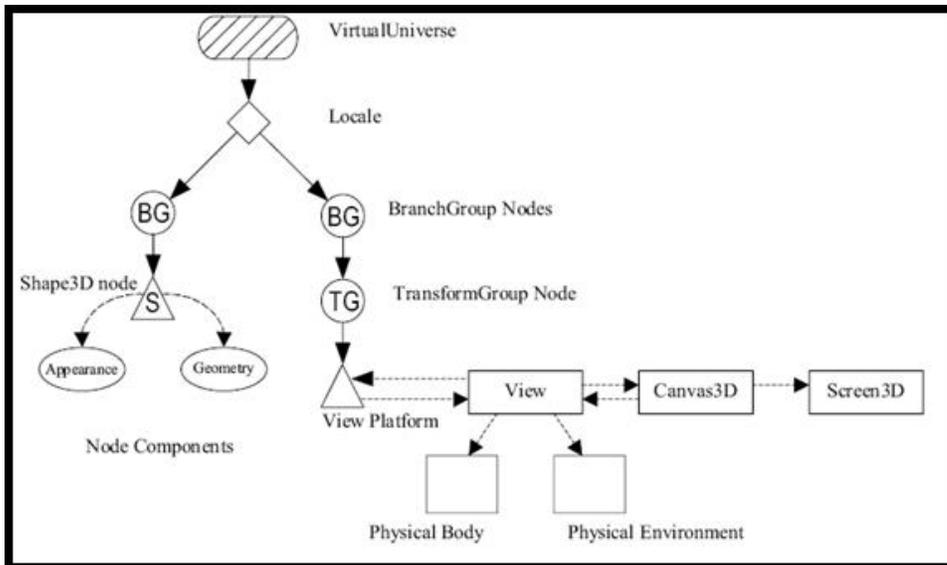


Figura 8 - Estrutura de um Grafo de cena (Sun 2010a)

Ainda segundo a autora supracitada, o grafo de cena é criado a partir de instancias de classe Java 3D, e define diversas características do grafo como som, orientação, geometria, aparência e localização dos objetos. Cada objeto corresponde a um nodo, também chamado de vértice, cada relacionamento seu e indicado pelo arco ou arestas, estas característica mostram ditam dois tipos de relacionamentos a referência que liga o objeto ao grafo de cena e a herança (pai-filho), onde um “nodo do tipo grupo” pode ter um ou mais filhos, mas apenas um pai, e um “nodo do tipo folha” não pode ter filhos.

Num grafo de cena, os nodos do tipo grupo são identificados por círculos, e do tipo folha por triângulos. Como ilustra a Figura 8, seguida da Figura 9 com suas legendas, os objetos são organizados em uma estrutura de árvore em um grafo de cena. Assim, um nodo é considerado raiz, e os demais são acessados seguindo os arcos a partir da raiz.

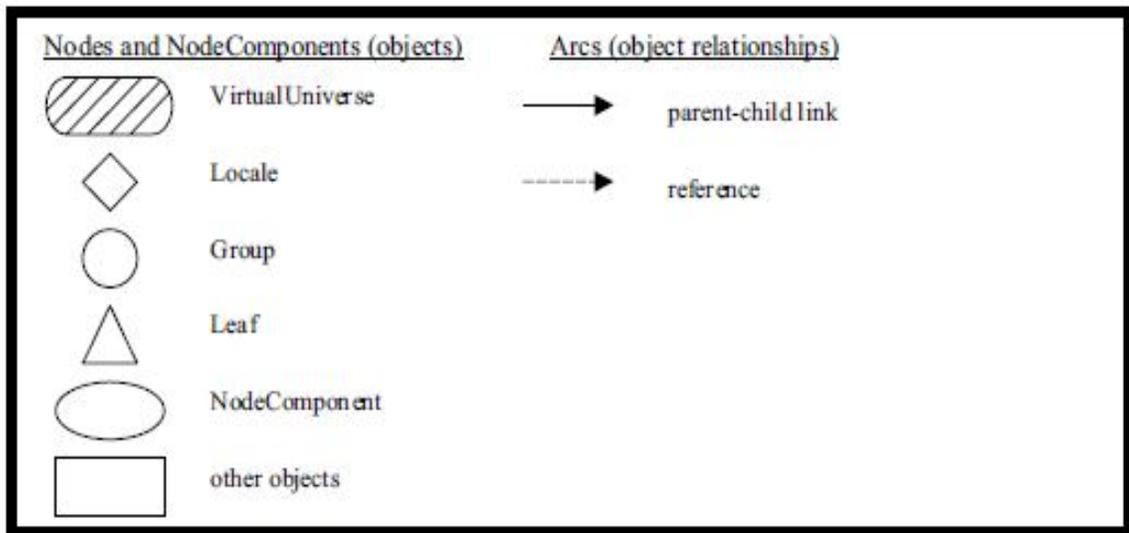


Figura 9 - Legenda do Grafo de Cena (SUN, 2010b)

Recomenda-se que cada grafo de cena possua um único objeto *VirtualUniverse*. Este objeto define um universo e possui pelo menos um objeto *Locale*, que é responsável pela especificação do ponto de referência no universo virtual e serve como raiz dos sub-grafos de um grafo de cena.

Na Figura 8, por exemplo, existem dois sub-grafos cujas raízes são objetos do tipo grupo (*BranchGroup*). O objetivo destes objetos é agrupar nodos relacionados através de alguma associação comum ou através de um conjunto de características. Existem duas categorias diferentes de sub-grafos que geralmente são incluídos em um grafo de cena. A primeira, representada pela sub-árvore da esquerda na Figura 8, descreve o conteúdo do universo virtual (*content branch graphs* ou sub-grafo de conteúdo), tais como geometrias, aparências, comportamentos, localizações, sons e luzes. A segunda, representada pela subárvore da direita na Figura 8, especifica os parâmetros de controle da visualização da cena (*view branch graphs* ou sub-grafo de visualização), tal como direção de observação.

Na Figura 8 se pode observar que todos os nodos do tipo grupo são representados graficamente por círculos, nodos folhas são identificados por triângulos e os demais objetos por retângulos. O nodo *TransformGroup* é usado para especificar a posição (relativa a *Locale*), orientação e escala dos objetos geométricos no universo virtual. Os nodos folha no exemplo da Figura 8 são *Behavior*, *Shape3D* e *ViewPlatform*. *Behavior* contém o código Java necessário para manipular a matriz de transformação associada com a geometria do objeto. *Shape3D* refere-se a dois objetos: *Geometry*, que fornece a forma geométrica do objeto, e *Appearance*, que descreve a aparência da geometria, isto é, sua cor, textura, características de reflexão, entre outras. Finalmente, *ViewPlatform* define a visão final do usuário dentro do universo.

9.1.4 - Carregando Modelos Geométricos Prontos

Segundo Barrilleaux (2001) *apud* Pavarini (2006) lembra que a API Java 3D não possui um formato de arquivo próprio, porém permitem desenvolver classes chamadas *Loaders*, também chamadas de carregadores. A classe *Loader* lê gráficos particulares, ou seja, formatos de arquivos que são instanciados como um objeto no grafo da cena. Em seus utilitários, a SUN criou a classe *LoaderBaser* que permite instanciar a classe *Loader* e uma interface para carregar os arquivos, esta classe que é definida no pacote *com.sun.j3d.loaders* permite a importação de uma grande

variedade de formatos de arquivos, estes arquivos estão descritos na Tabela 1 na página seguinte.

**Tabela 1 - Tipos de arquivos que podem ser carregados na API Java 3D.
(Sun 2010)**

Formato do Arquivo	Software que Gera o Arquivo
3DS	<i>3D – Studio</i>
COB	<i>Caligari trueSpace</i>
DEM	<i>Digital Elevation Map</i>
DXF	<i>AutoCAD Drawing interchange File</i>
IOB	<i>Imagine</i>
LWS	<i>LightWave Scene Format</i>
NFF	<i>WorldToolKit NFF Format</i>
OBJ	<i>Wavefront</i>
PDB	<i>Protein Data Bank</i>
PLAY	<i>PLAY</i>
SLD	<i>Solid Works (arquivos .prt e .asm)</i>
VRT	<i>Superscape</i>
VTK	<i>Visual Toolkit</i>
WRL	<i>Virtual Reality Modeling Language</i>
X3D	<i>Extended 3D</i>

As classes *Loader* e *Scene* da API Java3D definem interfaces que são comuns e totalmente responsáveis pelo carregamento de arquivos assim retornando um objeto na cena. O arquivo *wavefront*, que pode ser chamado de aramado, é um dos arquivos que são possíveis de ser carregados. Sua extensão é denominada como *.obj* como mostra a Tabela 1, este formato foi escolhido para importar os modelos 3D do protótipo porque tem uma estrutura relativamente simples.

Para a importação de arquivos usando a API Java 3D ocorrer é necessário utilizar algumas bibliotecas referentes à classe *Loader*, neste caso são 3 as bibliotecas *Loader*, sem elas não é possível carregar ou exportar objetos 3D para o Java, demonstrando isso que foi abordado à seguir o código fonte de como se faz isso na Figura 10 na página seguinte.

```
1
2 import java.applet.Applet;
3 import javax.swing.*;
4 import java.net.URL;
5 import java.awt.*;
6 import java.io.*;
7 import com.sun.j3d.utils.geometry.*;
8 import com.sun.j3d.utils.universe.*;
9 import com.sun.j3d.utils.behaviors.vp.*;
10 import com.sun.j3d.loaders.objectfile.ObjectFile;
11 import com.sun.j3d.loaders.ParseException;
12 import com.sun.j3d.loaders.IncorrectFormatException;
13 import com.sun.j3d.loaders.Scene;
14 import javax.media.j3d.*;
15 import javax.vecmath.*;
```

Figura 10 - Código para importação de bibliotecas para permitir o carregamento de objetos dentro da API Java 3D (SUN, 2010)

Depois de incluídas as bibliotecas são necessárias algumas linhas de códigos para que a importação do arquivo com extensão .obj ocorra, a figura 11(página seguinte) mostra um trecho de código responsável pelo carregamento do objeto 3D, nele é possível observar a linha *ObjectFile* onde ele cria um espaço para a figura ser carregada e usando o *try catch* carrega o objeto 3D e faz um tratamento de erros, caso eles ocorram no carregamento.

```
ObjectFile f =
    new ObjectFile(ObjectFile.RESIZE, (float) (60.0 * Math.PI / 180.0));
Scene s = null;

try {
    s = f.load(new java.net.URL(getCodeBase().toString() + "./mama.obj"));
} catch (FileNotFoundException e) {
    System.err.println(e);
    System.exit(1);
} catch (ParseException e) {
    System.err.println(e);
    System.exit(1);
} catch (IncorrectFormatException e) {
    System.err.println(e);
    System.exit(1);
} catch (java.net.MalformedURLException ex) {
    System.out.println(ex.getMessage());
    System.exit(1);
}

objRaiz.addChild(s.getSceneGroup());

objRaiz.compile();

return objRaiz;
```

Figura 11 - Código responsável pela importação ou carregamento do objeto 3D para a API Java 3D.

Feita a importação do objeto 3D é possível executar o código e visualizá-lo em uma cena utilizando a API Java 3D. Esta cena pode ser observada na Figura 12.

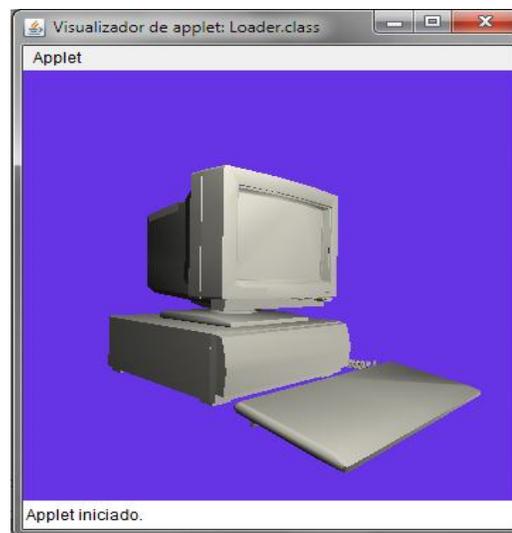


Figura 12 - Objeto 3D carregado na API Java 3D (MANSSOUR, 2003)

9.1.5 – Software de Modelagem Blender

Segundo Blender(2010) o software de modelagem Blender foi concebido em dezembro de 1993 e se tornou um produto utilizável em agosto de 1994 como uma aplicação integrada, que permite a criação de uma ampla gama de produtos 2D e 3D. O Blender fornece um amplo conjunto de recursos de modelagem 3D, texturização, iluminação, animação e pós-processamento de vídeo e desenvolvimento de jogos 3D, integrando todas essas funcionalidades em um único pacote.

Por meio de sua Arquitetura Aberta, Blender fornece interoperabilidade entre plataformas, é extensível, exigindo muito pouco do poder de processamento do computador, e um fluxo de trabalho totalmente integrado. Blender é um dos mais populares aplicativos Open Source²Gráfico 3D no mundo. Destinado a nível mundial á profissionais de mídia e artistas, Blender pode ser usado para criar visualizações 3D, bem como vídeos de qualidade para a TV e o cinema, enquanto que a incorporação de um motor 3D em tempo real permite a criação de conteúdo 3D interativo para a reprodução de *stand-alone*. Originalmente desenvolvido pela empresa 'Not a Number' (NaN), o Blender é agora "Software Livre", com código fonte disponível sob licença gratuita. Atualmente é desenvolvido pela *Blender Foundation* sediada na Holanda.

Suas características principais são que, possui uma variedade de ferramentas de criação, que são essenciais para a criação de conteúdo 3D, modelagem, incluindo mapeamento UV, texturização, *rigging*, *skinning*, animação, simulação de partículas, *scripts*, renderização, composição, física, pós-produção e criação de jogos.

Se trata de um Software de plataforma cruzada, com uma uniforme interface gráfica OpenGL em todas plataformas, pronta para todas as versões do Windows (98, NT, 2000, XP, Vista, 7), Linux, Mac OS X, FreeBSD, Irix , Sun e vários outros sistemas operacionais.

Possui uma arquitetura 3D de alta qualidade que permite a criação de um rápido e eficiente fluxo de produção, podendo ser executado ate mesmo de pen drives.

² Open Source: Software que tem distribuição gratuita.

10 – DESENVOLVIMENTO DA MODELAGEM E DA PROGRAMAÇÃO JAVA DO PROTÓTIPO

Neste capítulo será descrito o protótipo que tem como função visualizar uma mama em 3D assim auxiliando os profissionais da medicina e respectivamente seus pacientes a entender melhor o desenvolvimento e como se diagnostica e previne o câncer de mama. Será descrito o processo de modelagem da mama e o processo de criação do ambiente 3D.

10.1 – MODELAGEM DO OBJETO VIRTUAL

No início do projeto foi proposto que seria modelado no software Blender um avatar para ser usado no protótipo de prevenção do câncer de mama, porém no decorrer das pesquisas, verificou-se que como o Blender é um software complexo e sua utilização depende de conhecimentos específicos, não haveria tempo hábil para modelar o avatar, por essa razão concluiu-se que seria melhor utilizar somente a mama, sendo assim, a mama utilizada foi modelada no 3D Studio por PAVARINI (2006), este arquivo é do tipo *wavefront*³, assim podendo ser exportado para diversos softwares de modelagem, ele pode também ser chamado de aramado e tem a extensão .obj, o objeto 3D pode ser visto na figura 13 na folha seguinte (PALMER 2001).

³ Wavefront: tem uma estrutura relativamente simples permitindo a definição de malhas poligonais com associação de propriedades do material do objeto, segundo Palmer (2001).

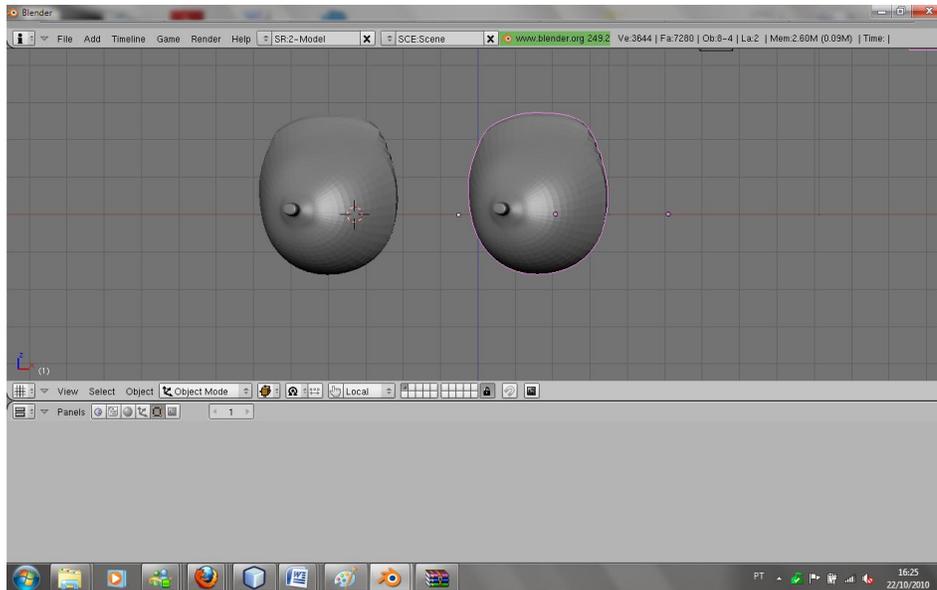


Figura 13 - Mamas que serão utilizadas no protótipo

10.2 – PROGRAMAÇÃO DO PROTÓTIPO

Como descrito anteriormente toda a parte de programação foi feita em Java utilizando a API Java 3D, a seguir será descrito todo o processo de montagem do código fonte.

No decorrer das pesquisas constatou-se que, havia diversos trabalhos disponíveis na internet de profissionais renomados da área e que poderiam ser utilizados como objeto de pesquisa para implementação do código fonte Java deste protótipo. O código principal que foi utilizado como pesquisa foi o feito por Manssour (2006). Das pesquisas feitas e de conhecimento da linguagem foram feitas adaptações no código fonte para carregamento de imagens 3D do tipo wavefront, e extensão .obj. O código completo será disponibilizado em anexo no final deste trabalho, neste capítulo será descrito trechos cruciais para o seu perfeito funcionamento.

Primeiramente é preciso falar sobre a escolha das bibliotecas, depois de pesquisas feitas em livros foi constatado a necessidade de bibliotecas específicas da API Java 3D para o devido funcionamento, já que o código carrega uma imagem

com extensão .obj foi necessário o uso das bibliotecas Loaders, que são específicas da API Java 3D e cujo significado foi explicado em capítulos anteriores. A sintaxe da biblioteca pode ser observada na figura 14.

```
import com.sun.j3d.loaders.objectfile.ObjectFile;  
import com.sun.j3d.loaders.ParsingErrorException;  
import com.sun.j3d.loaders.IncorrectFormatException;  
import com.sun.j3d.loaders.Scene;
```

Figura 14 - Bibliotecas loaders da API Java 3D

Há outras bibliotecas que foram utilizadas, seus nomes e funções estão descritos na figura 15.

Pacote API	Suporta
java.applet	Recursos gerais dos applets
java.awt	Recursos gráficos: botão, barra, caixas de texto, janela, etc. Em outras palavras: recursos GUI (Graphical User Interface).
java.io	Input e output de dados (entrada e saída)
java.lang	Recursos de linguagem
java.math	Operações matemáticas
java.net	Rede
java.security	Segurança
java.text	Recursos de texto
java.util	Miscelânea de recursos utilitários

Figura 15 - Principais bibliotecas utilizadas no protótipo (Manssour 2010)

Após a escolha das bibliotecas corretas, foi necessário pesquisar em relação a classe específica de carregamento que a API Java 3D necessita, foi implementada uma classe chamada *Loader* com um *extends Applet* assim adicionando funcionalidades a ela, no caso da applet, foi usado para possibilitar a exibição do objeto 3D.

Foi criado um método chamado *SimpleUniverse* e atribuído um *null* à ele, para certificar-se de que não havia nenhum lixo de memória armazenado neste método resultado de outras aplicações, neste método *SimpleUniverse* foi criada toda a cena do protótipo. Após isso foi feita toda a parte de criação do canvas.

O código seguinte alterou a *viewplatform* e fez com que ela fosse movida para trás, possibilitando a visualização do objeto que será carregado, junto com ele é alterado o campo de visão, que no código está descrito como *field-of-view*, tal modificação foi feita para que a visualização do objeto 3D por parte do usuário seja a melhor. Assim a base para que seja carregada a imagem 3D está pronta, o trecho do código pertinente pode ser visto na figura 16 na página seguinte, lembrando que o código completo estará em anexo no final deste trabalho.

```
19 public class Loader extends Applet {
20
21     private static class Enumeration {
22
23         public Enumeration() {
24             }
25     }
26     private SimpleUniverse universe = null;
27
28     @Override
29     public void init() {
30         setLayout(new BorderLayout());
31         GraphicsConfiguration config =
32             SimpleUniverse.getPreferredConfiguration();
33
34         Canvas3D canvas = new Canvas3D(config);
35         add("Center", canvas);
36
37
38         BranchGroup scene = criaGrafoDeCena();
39         universe = new SimpleUniverse(canvas);
40
41
42         ViewingPlatform viewingPlatform = universe.getViewingPlatform();
43         viewingPlatform.setNominalViewingTransform();
44
45
46         View view = universe.getViewer().getView();
47         view.setFieldOfView(view.getFieldOfView() * 1.4);
```

Figura 16 - Trecho inicial do código do protótipo

Após a criação do canvas foi necessário criar um *Behavior*, que seria um comportamento para o objeto carregado ,podendo assim ser manipulado através do mouse, por essa razão foi atribuído uma órbita e alguns limites, assim quando o usuário estiver interagindo com o objeto virtual através do mouse não poderá movê-lo para fora da cena. Mais abaixo o *universe.addBranchGroup*, anexa o sub-grafo no universo virtual, e com o “*public BranchGroup criaGrafoDeCena*”é criado o grafo de cena.Após isso foi criado o nodo *TransformGroup*, assim permitindo alterar o objeto em tempo de execução, em seguida foi adicionado ao grafo de cena. O código referente encontra-se na página seguinte na figura 17.

```
50 OrbitBehavior orbit =
51     new OrbitBehavior(canvas, OrbitBehavior.REVERSE_ALL);
52 BoundingSphere bounds =
53     new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);
54 orbit.setSchedulingBounds(bounds);
55 viewingPlatform.setViewPlatformBehavior(orbit);
56
57
58 universe.addBranchGraph(scene);
59
60 setSize(350, 350);
61 setVisible(true);
62 }
63
64 public BranchGroup criaGrafoDeCena() {
65
66
67     BranchGroup objRaiz = new BranchGroup();
68
69
70     TransformGroup objTrans = new TransformGroup();
71     objTrans.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);
72     objRaiz.addChild(objTrans);
73
74
75     BoundingSphere bounds =
76         new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);
77
78
```

Figura 17 - Trecho do código responsável pela criação do grafo de cena e de outros componentes.

Nas próximas duas figuras a seguir, será exposto trechos de códigos onde foi atribuída cores a componentes do protótipo, a figura 18 atribuiu uma cor ao

background que seria o fundo da cena, a cor é atribuída através do padrão RGB, que significa *Red, Green, Blue*, que traduzindo para o português significa vermelho, verde, azul, através da sincronia destas cores foi escolhida a cor azul para o fundo do protótipo. Já na figura 19 cria-se uma luz ambiente e atribuiu-se uma cor a ela, pelo mesmo padrão RGB

```
Color3f bgColor = new Color3f(0.4f, 0.2f, 0.9f);  
Background bg = new Background(bgColor);  
bg.setApplicationBounds(bounds);  
objRaiz.addChild(bg);
```

Figura 18 - Atribui cor ao background

```
Color3f corAmb = new Color3f(0.2f, 0.2f, 0.2f);  
AmbientLight luzAmb = new AmbientLight(corAmb);  
luzAmb.setInfluencingBounds(bounds);  
objRaiz.addChild(luzAmb);
```

Figura 19 0- Cria e atribui cor à luz ambiente

Para uma perfeita visualização do objeto 3D foi necessário a criação de pontos de luz independentes, assim iluminando todos os pontos do objeto 3D, já que tal objeto será rotacionado e transladado. O posicionamento dos pontos de luz é atribuído através de coordenadas espaciais, onde o primeiro valor é o x, que representa largura, o segundo y representado altura e o terceiro z representado profundidade, os três eixos serão esquematizadas na figura 20 e o código correspondente está descrito na figura 21, ambas figuras na página seguinte.

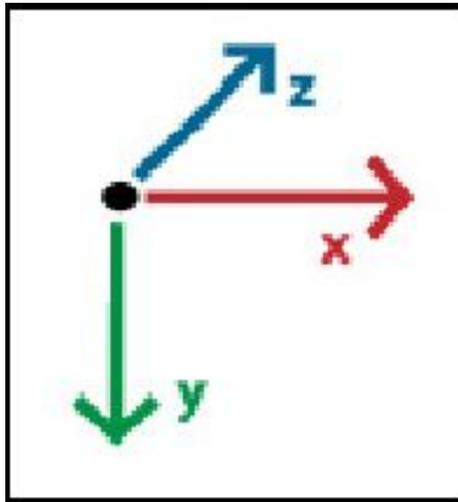


Figura 20 - Os três eixos usado para posicionamento dos pontos de luz

```
Color3f corLuz = new Color3f(0.9f, 0.9f, 0.9f);
Point3f posicaoLuz1 = new Point3f(2.0f, 2.0f, 1.2f);

Point3f posicaoLuz2 = new Point3f(2.0f, -2.0f, -2.0f);

Point3f posicaoLuz3 = new Point3f(3.0f, 3.0f, 3.0f);

Point3f posicaoLuz4 = new Point3f(-3.0f, -3.0f, -3.0f);

Point3f atenuacaoLuz = new Point3f(0.1f, 0.1f, 0.1f);

PointLight luzPont = new PointLight(corLuz, posicaoLuz1, atenuacaoLuz);
luzPont.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont);

PointLight luzPont2 = new PointLight(corLuz, posicaoLuz2, atenuacaoLuz);
luzPont2.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont2);

PointLight luzPont3 = new PointLight(corLuz, posicaoLuz3, atenuacaoLuz);
luzPont3.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont3);

PointLight luzPont4 = new PointLight(corLuz, posicaoLuz4, atenuacaoLuz);
luzPont3.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont4);
```

Figura 21 - Criação dos pontos de luzes e posicionamento

Depois de criado o univeros virtua, grafo de cena, sub_grafo de cena e iluminação, foi feito o código para carregar o arquivo com extensão .obj.

Primeiramente foi chamada a função ObjectFile para que haja um suporte para o carregamento da imagem 3D, logo após foi atribuido null a cena garantindo que não fique nenhum lixo de memória de outras aplicações, em seguida é implementado o try que tem como significado no java de tentar, em seguida é passado é chamada a biblioteca "java.net.URL que tem como função chamar arquivos que estão armazenados em um computador local, após isso foi escrito o caminho que foi salvo a imagem 3d, a imagem tem que ser salva no diretório do projeto, como neste caso a IDE utilizada é o NetBeans ela tem que estar salva dentro do caminho que mostra a figura 22

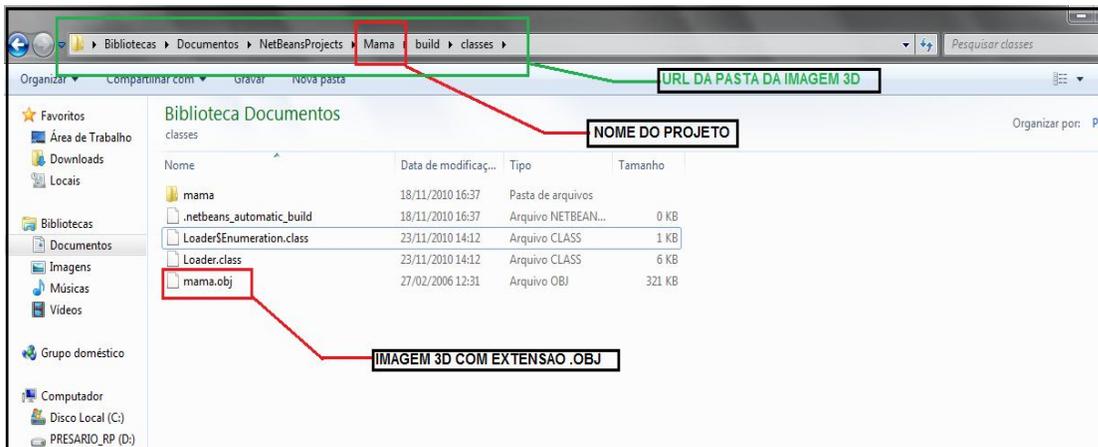


Figura 22 - URL onde deve ser armazenado o arquivo com extensão .obj

Após a URL correta foi digitada no código, possibilitando o devido carregamento do objeto 3D, é feito um tratamento de erros com o Catch, ou seja se durante o carregamento do arquivo 3D ocorrer algum erro como arquivo corrompido , ou algo parecido não travará o aplicativo e sim exibirá uma mensagem do que está ocorrendo. No final do código forma feitas otimizações no grafo de cena para que funcione de melhor maneira. Trecho de código referente as descrições acima pode ser observado na figura 23 na página seguinte.

```

ObjectFile f =
    new ObjectFile(ObjectFile.RESIZE, (float) (60.0 * Math.PI / 180.0));
Scene s = null;

try {
    s = f.load(new java.net.URL(getCodeBase().toString() + "/mama.obj"));
} catch (FileNotFoundException e) {
    System.err.println(e);
    System.exit(1);
} catch (ParsingErrorException e) {
    System.err.println(e);
    System.exit(1);
} catch (IncorrectFormatException e) {
    System.err.println(e);
    System.exit(1);
} catch (java.net.MalformedURLException ex) {
    System.out.println(ex.getMessage());
    System.exit(1);
}

objRaiz.addChild(s.getSceneGroup());
objRaiz.compile();

return objRaiz;
}
}

```

Figura 23 - Trecho de código onde é carregado o arquivo com extensão .obj

11 – FUNCIONAMENTO DO PROTÓTIPO PREVINE CÂNCER 2010

O Previne Câncer 2010 como descrito anteriormente dispõe de um visualizador da mama e oferecerá à usuária maneiras de interagir com a mama, podendo rotacionar o órgão 360° em todas as direções e transladar para qualquer área da tela do protótipo. Seu público alvo serão os pacientes portadores ou não do câncer de mama, que estejam em tratamento ou prevenção da doença.

Dará à usuária com câncer de mama uma maneira de conhecer mais o órgão onde é desenvolvido a doença.

Uma breve visão do Previne Câncer 2010 pode ser vista na figura 24.

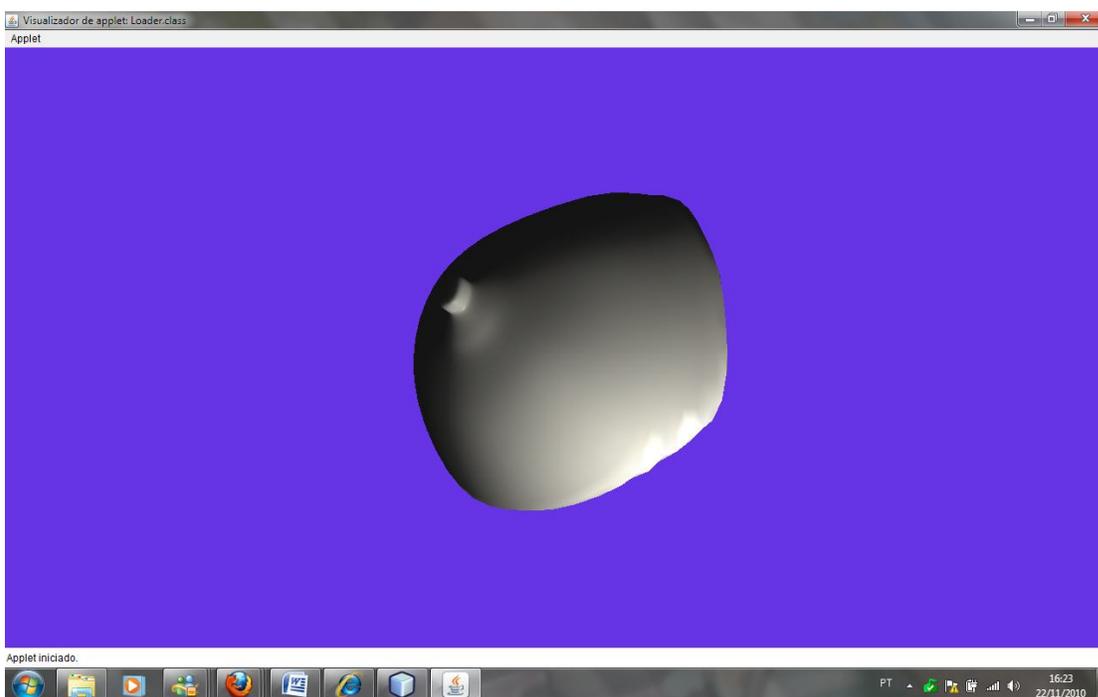


Figura 24 - Tela inicial do protótipo, com a imagem 3d de uma mama

12 – CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

O presente trabalho apresentou um protótipo de um aplicativo em realidade virtual chamado Previne Câncer 2010. Apesar de nem todos os objetivos propostos inicialmente não terem sido alcançados, durante as pesquisas e o desenvolvimento percebeu-se que o protótipo será de grande importância para as mulheres com ou sem câncer de mama, o protótipo é capaz de dar ao usuário uma excelente visualização do órgão feminino, constatou-se também através de pesquisas descritas nos capítulos deste trabalho, que uma doença com uma alta taxa de incidência necessita de ferramentas que auxiliam na prevenção e no tratamento.

No decorrer do trabalho foi possível notar como desenvolver ambientes 3D para auxiliar na área da medicina, também foi possível constatar que através do Previne Câncer 2010 e de futuras adaptações o software poderá atuar como auxílio em tratamentos médicos, programas de prevenção e conscientização de diversas doenças existentes no mundo inteiro, pois tanto o tipo de modelagem, quanto a linguagem Java são conhecidas e trabalhadas mundialmente, todas essas opções podem ser possíveis devido a possibilidade de troca do objeto 3D do tipo wavefront com extensão .obj, lembrando que pode ser modelado em qualquer software de modelagem 3d, só sendo necessária a gravação em extensão .obj, assim ele pode se tornar importável por qualquer software e principalmente pela API Java 3D.

Como trabalhos futuros poderiam ser implementadas as funções propostas inicialmente e que não foram possíveis serem implementadas por se tratarem de funções que dependiam de conhecimentos específicos em medicina, ou seja necessitaria um acompanhamento médico, no caso da demonstração dos tipos de câncer um especialista na doença tratada, na prevenção e na interação do software com a pessoa com câncer seria necessário acompanhamento de um psicólogo para se ter a correta abordagem do assunto com o paciente, só através desses acompanhamentos seria possível a implementação desta parte do projeto. As funções que serão implementadas futuramente são: visualização do câncer de mama em diferentes estágios de crescimento, visualização de diferentes tipos de cânceres, e simulação do auto – exame.

REFERÊNCIAS BIBLIOGRÁFICAS

BLEZEK, D. J.; ROBB R. A.; MARTIN, D. P. Virtual Reality Simulation of Regional Anesthesia for Training of Residents. In: HICSS 33rd Hawaii international Conference on System Sciences, 2000. **Proceedings...**Washington, DC: IEEE Computer Society, Vol 5, p.5022.

DEITEL, H.M. e DEITEL, P.J. **Java Como Programar**. 8º ed. São Paulo:Pearson Prentice Hall, 2010. 1144p.

FENTIMAN, I. Diagnóstico e Tratamento do Câncer Inicial de Mama. Porto Alegre:Artes Médicas, 1993.

INCA.(2010)**Câncer de Mama**. Disponível em:<http://www2.inca.gov.br/wps/wcm/connect/tiposdecancer/site/home/mama..> Acesso em: jun. 2010.

JACOBSON, L. **Virtual Reality: A status Report**. Pp. 26 – 33 AI Expert. Agosto, 1991.

JACOBSON, L. **Realidade Virtual em Casa**. Rio de Janeiro: Berkeley, 1994.

KIRNER, C. **Apostila do Ciclo de Palestras de Realidade Virtual.Atividade do Pojeto AVVIC-CNPq** (Protem – CC – faseIII) – DC/UFSCar, pp. 1-10. São Carlos, Outubro, 1996.

LIU, A.; TENDICK, F.; CLEARY, K.; KAUFMANN, C. .A Survey of Surgical Simulation: Application, Technology and Education. In: MIT Press, vol. 12, Dezembro, 2003.

MACHADO, L. S.; CAMPOS, S. F; CUNHA, Í. L; MORAES, R. M. de. **Cybermed: Realidade Virtual Para Ensino Médico**. In: III CONGRESSO LATINO-AMERICANO DE ENGENHARIA BIOMÉDICA, 2004, João Pessoa. **Proceedings...** p. 573-576.

MACHADO, L. S. A Realidade Virtual no Modelamento e Simulação de Procedimentos **Invasivos em Oncologia Pediátrica: Um Estudo de Caso no Transplante de Medula Óssea**. 2003, 130 f. Grau: Tese (Doutorado em Engenharia) Departamento de Engenharia de Sistema Eletrônicos da Escola Politécnica da Universidade de São Paulo, 2003. Disponível em: < <http://www.teses.usp.br/teses/disponiveis/3/3142/tde-07052003-123257/> > . Acesso em: 21 jun. 2010.

MANSSOUR.(2010)**Introdução à Java 3D**. Disponível em: <http://www.inf.pucrs.br/~manssour/Java3D/> Acesso em: Set. 2010.

MONTGOMERY, K.; HEINRICHS, L.; BRUYNS, C.; WILDERMUTH, S.; HASSER, C.; OZENNE, S.; BAILEY, D. Surgical Simulator for Hysteroscopic: A Case Study of

Visualization in Surgical Training. In: 12th IEEE VISUALIZATION CONFERENCE, 2001 San Diego-CA. **Proceedings...**

MONTEIRO, B. (2006). **Implementação de Módulo de Avaliação da Aprendizagem em uma Ferramenta Virtual para Treinamento Médico**. 2006. 99 f. Grau: Monografia (Bacharelado em Ciência da Computação) Centro Universitário Eurípides de Marília –Fundação de Ensino Eurípides Soares da Rocha, Marília, 2006.

NETTO, A. V., MACHADO, L. S., OLIVEIRA, M. C .F. **Realidade Virtual: Fundamentos e Aplicações**. Florianópolis: Visual Books, Agosto, 2002.

NETTO, A V.; MACHADO, L.; OLIVEIRA, M.C.F. **Realidade Virtual – Definições, Dispositivos e Aplicações**. Revista Eletrônica de Iniciação Científica. Publicação da Sociedade Brasileira de Computação, Vol 2, No. 1, 2002.

NINISS, H.; INOUE, T. Electric Wheelchair Simulator for Rehabilitation of Persons with Motor Disability. In: SVR 2006 – VIII SYMPOSIUM ON VIRTUAL REALITY, 2006, Belém. **Proceedings...**Belém: CESUPA, 2006, p.51-62.

PAVARINI, L., NUNES, F. L. S., OLIVEIRA, A. C. T. G., BOTEAGA, L. C., BEZERRA, A. **DefApliMed – Sistema de Deformação para Aplicações Médicas com base no método Massa-Mola utilizando a API Java 3D**. Dissertação de Mestrado, Universitário Eurípides de Marília, Marília – SP – Brasil.

PALMER, I. **Essential Java 3D Fast - Developing 3D Graphics Applications in Java**. 1º ed. SPRINGER VERLAG NY, 2001, 279p.

PUCRS(2006) **INTRODUÇÃO A JAVA 3D**. disponível em: <http://www.inf.pucrs.br/~manssour/Publicacoes/TutorialSib2003.pdf>. Acesso em:novembro. 2010

SUN.(2010). **Java 3D API Tutorial**. Disponível em: < <http://java.sun.com/developer/onlineTraining/java3d/>> . Acesso em: jun. 2010

SUN(2010). **Getting Started with the Java 3D API**. Disponível em: http://java.sun.com/developer/onlineTraining/java3d/j3d_tutorial_ch1.pdf. Acesso em: nov. 2010

WEBSTER, R.; ZIMMERMAN, D.; MOHLER, B.; MELKONIAN, M.; HALUCK, R.A **Prototype Haptic Suturing Simulator**. In: Medicine Meets Virtual Reality - Studies in Health Technology and Informatics,2001. IOS Press, n. 81, p. 567-569.

ANEXO

Programação do Ambiente e do Objeto 3D

Loader.java

```
import java.applet.Applet;
import javax.swing.*;
import java.net.URL;
import java.awt.*;
import java.io.*;
import com.sun.j3d.utils.geometry.*;
import com.sun.j3d.utils.universe.*;
import com.sun.j3d.utils.behaviors.vp.*;
import com.sun.j3d.loaders.objectfile.ObjectFile;
import com.sun.j3d.loaders.ParsingErrorException;
import com.sun.j3d.loaders.IncorrectFormatException;
import com.sun.j3d.loaders.Scene;
import javax.media.j3d.*;
import javax.vecmath.*;

public class Loader extends Applet {

    private static class Enumeration {

        public Enumeration() {
        }
    }

    private SimpleUniverse universe = null;

    @Override
    public void init() {
        setLayout(new BorderLayout());
    }
}
```

```

GraphicsConfiguration config =
    SimpleUniverse.getPreferredConfiguration();

Canvas3D canvas = new Canvas3D(config);
add("Center", canvas);

BranchGroup scene = criaGrafoDeCena();
universe = new SimpleUniverse(canvas);

ViewingPlatform viewingPlatform = universe.getViewingPlatform();
viewingPlatform.setNominalViewingTransform();

View view = universe.getViewer().getView();
view.setFieldOfView(view.getFieldOfView() * 1.4);

OrbitBehavior orbit =
    new OrbitBehavior(canvas, OrbitBehavior.REVERSE_ALL);
BoundingSphere bounds =
    new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);
orbit.setSchedulingBounds(bounds);
viewingPlatform.setViewPlatformBehavior(orbit);

universe.addBranchGraph(scene);

setSize(350, 350);
setVisible(true);
}

```

```
public BranchGroup criaGrafoDeCena() {  
  
    BranchGroup objRaiz = new BranchGroup();  
  
    TransformGroup objTrans = new TransformGroup();  
    objTrans.setCapability(TransformGroup.ALLOW_TRANSFORM_WRITE);  
    objRaiz.addChild(objTrans);  
  
    BoundingSphere bounds =  
        new BoundingSphere(new Point3d(0.0, 0.0, 0.0), 100.0);  
  
    Color3f bgColor = new Color3f(0.4f, 0.2f, 0.9f);  
    Background bg = new Background(bgColor);  
    bg.setApplicationBounds(bounds);  
    objRaiz.addChild(bg);  
  
    Color3f corAmb = new Color3f(0.2f, 0.2f, 0.2f);  
    AmbientLight luzAmb = new AmbientLight(corAmb);  
    luzAmb.setInfluencingBounds(bounds);  
    objRaiz.addChild(luzAmb);  
  
    Color3f corLuz = new Color3f(0.9f, 0.9f, 0.9f);  
    Point3f posicaoLuz1 = new Point3f(2.0f, 2.0f, 1.2f);  
  
    Point3f posicaoLuz2 = new Point3f(2.0f, -2.0f, -2.0f);
```

```

Point3f posicaoLuz3 = new Point3f(3.0f, 3.0f, 3.0f);

Point3f posicaoLuz4 = new Point3f(-3.0f, -3.0f, -3.0f);

Point3f atenuacaoLuz = new Point3f(0.1f, 0.1f, 0.1f);

PointLight luzPont = new PointLight(corLuz, posicaoLuz1, atenuacaoLuz);
luzPont.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont);

PointLight luzPont2 = new PointLight(corLuz, posicaoLuz2,
atenuacaoLuz);
luzPont2.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont2);

PointLight luzPont3 = new PointLight(corLuz, posicaoLuz3,
atenuacaoLuz);
luzPont3.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont3);

PointLight luzPont4 = new PointLight(corLuz, posicaoLuz4,
atenuacaoLuz);
luzPont3.setInfluencingBounds(bounds);
objRaiz.addChild(luzPont4);

ObjectFile f =
    new ObjectFile(ObjectFile.RESIZE, (float) (60.0 * Math.PI / 180.0));
Scene s = null;

try {
    s = f.load(new java.net.URL(getCodeBase().toString() + "./mama.obj"));

```

```
    } catch (FileNotFoundException e) {
        System.err.println(e);
        System.exit(1);
    } catch (ParseException e) {
        System.err.println(e);
        System.exit(1);
    } catch (IncorrectFormatException e) {
        System.err.println(e);
        System.exit(1);
    } catch (java.net.MalformedURLException ex) {
        System.out.println(ex.getMessage());
        System.exit(1);
    }

    objRaiz.addChild(s.getSceneGroup());

    objRaiz.compile();

    return objRaiz;

}
}
```