



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Uso de Interação Natural em Simulador de Procedimentos para Treinamento no Setor Elétrico

Rafael Henrique Assis de Castro

João Pessoa - PB

Junho de 2015

UNIVERSIDADE FEDERAL DA PARAÍBA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Rafael Henrique Assis de Castro

**Uso de Interação Natural em Simulador de Procedimentos
para Treinamento no Setor Elétrico**

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal da Paraíba como parte dos requisitos necessários para obtenção do título de Mestre em Ciência da Computação.

Orientadores: Prof. Dr. Iguatemi Eduardo da Fonseca, Prof^a. Dra. Tatiana Aires Tavares

**João Pessoa – PB
Junho de 2015**

C355u Castro, Rafael Henrique Assis de.
 Uso de interação natural em simulador de procedimentos
 para treinamento no setor elétrico / Rafael Henrique Assis de
 Castro.- João Pessoa, 2015.
 121f. : il.
 Orientadores: Iguatemi Eduardo da Fonseca, Tatiana Aires
 Tavares
 Dissertação (Mestrado) - UFPB/CI
 1. Informática. 2. Ciência da computação. 3. Interação
 natural. 4. Simuladores de procedimentos. 5. Jogos -
 desenvolvimento. 6. Interação gestual.

UFPB/BC

CDU: 004(043)

UNIVERSIDADE FEDERAL DA PARAÍBA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

A banca examinadora, abaixo assinada, aprova a dissertação

**Uso de Interação Natural em Simulador de Procedimentos
para Treinamento no Setor Elétrico**

Elaborada por

Rafael Henrique Assis de Castro

Como requisito para obtenção do grau de

Mestre em Informática

Banca Examinadora

Prof. Dr. Iguatemi Eduardo da Fonseca (Presidente, Orientador)

Prof^a. Dr^a. Tatiana Aires Tavares (Orientadora)

Prof. Dr. Tarso Vilela Ferreira (Externo)

Prof. Dr. Claurton de Albuquerque Siebra (Interno)

João Pessoa – PB

23 de Junho de 2015

RESUMO

Simuladores de procedimentos têm sido utilizados em diversas áreas para auxiliar a execução de diversas tarefas. O uso de interfaces que facilitem a manipulação das ações, nas diversas fases da execução da tarefa, no contexto do simulador, pode ser um importante componente para facilitar o uso por partes dos profissionais. Sendo o setor elétrico composto por diversos procedimentos de alto risco, há uma grande necessidade do bom treinamento dos seus operadores. Visando auxiliar esse treinamento, o trabalho aqui descrito tem como principal objetivo projetar e executar a integração de interação natural em simuladores de procedimentos para o setor elétrico, realizando testes com voluntários do setor. Como estudo de caso, foram analisados cenários já desenvolvidos no simulador ATreVEE 3D e desenvolvida uma nova versão do simulador, chamada ATreVEE IN, dessa vez usando o controle Leap Motion, um dispositivo usado para o rastreamento das mãos. O ATreVEE IN é composto por ambientes virtuais cuja interface com o usuário é realizada via comandos gestuais, utilizando uma representação virtual das mãos do usuário dentro do ambiente. Após o desenvolvimento da nova versão, os dois simuladores foram testados por usuários da área de engenharia elétrica, utilizando a aplicação de questionários para a coleta de dados. Com esses resultados foi possível observar que ambas as versões tiveram desempenho satisfatório, mostrando um potencial para auxiliar o treinamento. E apesar da primeira versão do simulador ter obtido resultados melhores que a versão com interação natural, acredita-se que isso se dá, em parte, pela falta de experiência dos usuários com esse tipo de dispositivo. Usuários que já possuíam experiência com interação gestual obtiveram melhor desempenho na execução das tarefas com o ATreVEE IN do que os demais usuários.

Palavras-chave: Interação Natural, Simuladores de Procedimentos, Desenvolvimento de Jogos, Interação Gestual.

ABSTRACT

Procedure Simulators have been used in several areas to help the execution of various tasks. The use of an interface that facilitates the manipulation of the actions at different stages of execution of a task, in the context of the simulator, can be an important component to ease the use by the professionals. As the electricity sector consists of several high-risk procedures, there is a great need for the good training of their operators. Seeking to support this training, the work described here aims, design and implement the integration of natural interaction in procedure simulators for the electricity sector, conducting tests with volunteers of the sector. As a case study, scenarios already developed in ATreVEE 3D simulator were analyzed and a new version of the simulator was developed, called ATreVEE IN, this time using the Leap Motion Controller, a device used for tracking hands. The ATreVEE IN consists of virtual environments whose user interface is performed via gestural commands, using a virtual representation of the user's hands within the environment. After the development of the new version, the two simulators were tested by users of electrical engineering, using the application of questionnaires to collect data. With these data it was observed that both versions had satisfactory performance, showing a potential to aid training. And although the first version of the simulator have obtained better results than the version with natural interaction, it is believed that this occurs in part because of the lack of user experience with this type of device. Users who already have experience with gestural interaction performed better in executing the tasks with ATreVEE IN than other users.

Key-words: Natural Interaction, Procedure Simulator, Game Development, Gestural Interaction.

Lista de Figuras

Figura 1 - Realidade virtual não-imersiva e imersiva.	17
Figura 2 – Geomagic® Touch™	19
Figura 3 – Câmera ToF SwissRanger™ SR4000.	20
Figura 4 - Kinect para XBOX.	20
Figura 5 - Funcionamento do Kinect.	21
Figura 6 – Myo.....	23
Figura 7 - Funcionamento do Myo.	23
Figura 8 - Controle Leap Motion.	24
Figura 9 - Hardware do controle Leap Motion.	24
Figura 10 - Campo de visualização do Leap Motion.	25
Figura 11 - <i>Skeleton</i> das mãos.	25
Figura 12 - Isoladores usados e, linhas de 13,8 KV.....	38
Figura 13 - Poste com Isolador.	38
Figura 14 - Modelo proposto para desenvolvimento de simuladores de procedimentos.	40
Figura 15 - Cena de escolha de equipamentos no ATreVEE 3D.	43
Figura 16 - Escolha da ação com o cursor.	44
Figura 17 - Aplicação piloto.	46
Figura 18 - Controles da aplicação piloto: (a) visão superior do controle Leap Motion; (b) visão lateral do uso do controle Leap Motion; (c) gesto utilizado para pegar os objetos na cena.	47
Figura 19 – Máquina de estados para a troca do 1º isolador em linha ativa, com os passos desnecessários para a linha desativada destacados em verde.	48
Figura 20 – Diagrama de casos de uso para a manutenção em linha ativa.	49
Figura 21 - Interface do editor do Unity 3D.	50
Figura 22 - Ossos da mão.....	51
Figura 23 - Arquitetura do ATreVEE IN.	52
Figura 24 - Diagrama de pacotes para cena de transição.....	53
Figura 25 - Diagrama de pacotes para cena de preparação.....	54
Figura 26 - Diagrama de pacotes para cena de manutenção.....	54
Figura 27 - Cena de manutenção ATreVEE 3D.	55
Figura 28 - Cena de manutenção ATreVEE IN.	56
Figura 29 - Gerenciador de entrada e eixos horizontal e vertical.	57

Figura 30 - Código fonte da definição da direção de movimento com o teclado.	58
Figura 31 - Código fonte da definição da direção de movimento com o controle Leap Motion.	59
Figura 32 - Código fonte da definição da direção do vetor de direção com o controle Leap Motion.....	59
Figura 33 - Sistema de coordenadas do dispositivo Leap Motion (a) e do ambiente virtual (b).	60
Figura 34 - Movimento da personagem: (a) representação do movimento das mãos; (b) movimento resultante.....	60
Figura 35 - Código fonte da rotação da personagem com o mouse.....	61
Figura 36 - Código fonte da rotação da personagem com o controle Leap Motion.	61
Figura 37 - Código fonte da função que define o ângulo de rotação a partir do controle Leap Motion.....	62
Figura 38 - Classes definidas para os objetos interativos e suas relações.	63
Figura 39 - Código fonte das funções <i>AoPegar</i> e <i>PegarObjeto</i> da classe <i>Objeto</i>	64
Figura 40 - Código Fonte da função <i>Anexar</i> da classe <i>DestinoObj</i>	65
Figura 41 - Parte do código fonte: (a) da ferramenta correta no ATreVEE 3D; (b) da ferramenta errada no ATreVEE 3D.	66
Figura 42 - Parte do código fonte: (a) da ferramenta correta no ATreVEE IN; (b) da ferramenta errada no ATreVEE IN.....	67
Figura 43 – Modelos dos protetores usados: (a) canaleta protetora (no condutor); (b) cobertura isolante (na cruzeta).	68
Figura 44 - Modelo usado do isolador.	68
Figura 45 - Movimentação do condutor a partir do isolador.	69
Figura 46 - Modelo usado para o alicate.....	70
Figura 47 - Modelo usado para o fio de cobre.	72
Figura 48 - Obtenção dos valores da rotação de osso do dedo usando o SDK.....	73
Figura 49 - Atualiza rotação dos ossos do esqueleto do modelo 3D.	74
Figura 50 - Atualiza rotação e posição do modelo 3D da mão.....	74
Figura 51 - Modelos de mão usados: (a) modelo gráfico; (b) modelo físico.....	74
Figura 52 - Função que verifica ação de Fechar e Abrir.....	75
Figura 53 - Busca do objeto mais próximo da mão.	77
Figura 54 - Seleção de objeto mais próximo.	78
Figura 55 - Encerramento da cena da troca do primeiro isolador.....	79

Figura 56 - Gerenciamento do botão gráfico do isolador na troca do primeiro isolador.	80
Figura 57 - Estados usados na cena de manutenção.	80
Figura 58 - Ações de cena do gerenciador de cena de manutenção em linha ativa.	81
Figura 59 - Trecho de código que verifica se o primeiro isolador foi trocado.	82
Figura 60 - Cálculo da pontuação.	82
Figura 61 - Movimento do personagem com o mouse.	84
Figura 62 - Movimento da personagem com o controle Leap Motion.	84
Figura 63 - Cenário de transição para o cenário de manutenção: (a) interação com teclado e mouse; (b) interação com o controle Leap Motion.	85
Figura 64 - Cenário de preparação.	86
Figura 65 - Tela de pontuação.	86
Figura 66 - Cenários de Manutenção: (a) Linha ativa; (b) Linha desativada.	87
Figura 67 - Pontuação do Cenário de Manutenção.	87
Figura 68 - Etapas dos testes com usuários.	88
Figura 69 - Cenário usado para a Etapa 1.	89
Figura 70 - Demonstração do procedimento para os testes (Etapa 1).	91
Figura 71 - Teste com o ATreVEE 3D (Etapa 3).	92
Figura 72 - Teste com o ATreVEE IN (Etapa 5).	92
Figura 73 - Resultados das questões relacionadas com o aprendizado.	94
Figura 74 - Resultados das questões relacionadas com os dispositivos.	95
Figura 75 - Comparação de resultados de usuários com e sem experiência com interação gestual.	96

Lista de Tabelas

Tabela 1 - Relatório de Estatística de Acidentes do Setor Elétrico Brasileiro – 2012.....	14
Tabela 2 – Uso potencial dos SRVs.....	17
Tabela 3 - Comparativo de trabalhos.....	35
Tabela 4 - Passos para troca do primeiro isolador em linha ativa.	39
Tabela 5 - Passos para a execução dos procedimentos.....	42
Tabela 6 - Relações entre destinos e objetos.	65
Tabela 7 - Ações executadas no Fio de Cobre.....	70
Tabela 8 - Casos de objetos próximos válidos.....	76
Tabela 9 - Quantidade de passos para cada troca de isolador nas linhas ativa e desativada....	83

Lista de Anexos

ANEXO A - Formulário Usado Antes da Execução do Simulador.....	109
ANEXO B - Formulário Usado Após a Execução do Simulador com Mouse.	111
ANEXO C - Formulário Usado Após a Execução do Simulador com o Controle Leap Motion.	114
ANEXO D - Termo de Consentimento Assinado pelos Participantes dos Testes.	117
ANEXO E - Respostas do Questionário Pré-teste.	118
ANEXO F - Respostas do Questionário do Uso do ATreVEE 3D.....	119
ANEXO G - Respostas do Questionário do Uso do ATreVEE IN.....	120

Sumário

1. Introdução	13
1.1. Contextualização	13
1.2. Objetivos	14
1.3. Organização do Trabalho	15
2. Fundamentação Teórica	16
2.1. Realidade Virtual.....	16
2.2. Interação Natural	17
2.2.1. Meios de Interação Natural.....	18
2.2.1.1. Telas Sensíveis ao Toque	19
2.2.1.2. Sistemas Hápticos.....	19
2.2.1.3. Reconhecimento de Gestos/ Movimentos	19
2.2.1.4. Reconhecimento de Fala	22
2.2.1.5. Sistemas Controlados pelo Cérebro	22
2.2.1.6. Impulsos Elétricos Musculares.....	22
2.3. Controle Leap Motion	23
3. Simuladores de Treinamento	26
3.1. Treinamento com Realidade Virtual	27
3.2. Treinamento no Setor Elétrico	29
3.3. Treinamento com Interação Natural.....	33
3.4. Comparativo	34
4. ATreVEE 3D	37
4.1. Esquema Proposto para Simuladores de Procedimento	39
4.1.1. Cenário de Treinamento.....	41
4.1.1.1. Modos de Treinamento.....	41
4.1.1.2. Hierarquia de Operações	41
4.2. Cenários e Interação com o Usuário	42

5. ATreVEE IN	45
5.1. Aplicação Piloto	45
5.2. Funcionalidades.....	47
5.3. Desenvolvimento.....	49
5.3.1. Ferramentas Utilizadas.....	49
5.3.1.1. Unity 3D	49
5.3.1.2. Leap Motion SDK	50
5.3.2. Arquitetura	51
5.3.3. Construção da Cena	55
5.3.4. Navegação.....	57
5.3.5. Objetos Interativos	62
5.3.5.1. Objeto Base.....	63
5.3.5.2. Objeto	63
5.3.5.3. Destino de Objeto	64
5.3.5.4. Objetos da Cena de Preparação	65
5.3.5.5. Objeto Segurável e Objeto Interativo	67
5.3.5.6. Ferramenta	69
5.3.5.7. Objetos Usados com Ferramenta	70
5.3.5.8. Mãos	72
5.3.5.9. Controlador das Mãos.....	72
5.3.5.10. Gerenciadores de Jogo e Cena.....	78
5.3.5.11. Avaliação	82
5.4. Interação e Controle Leap Motion	83
5.5. Cenários.....	85
5.5.1. Cenários de Transição.....	85
5.5.2. Cenário de Preparação	85
5.5.3. Cenários de Manutenção.....	86

6. Testes com Usuários e Análise dos Resultados	88
6.1. Método	88
6.2. Material	88
6.3. Experimento	91
6.4. Resultados	92
7. Considerações Finais	98
7.1. Conclusão	98
7.2. Contribuições	100
7.3. Trabalhos Futuros.....	101
Referências.....	102
ANEXOS	109

Capítulo

1

Introdução

1.1. Contextualização

Com o avanço tecnológico e o aumento do acesso a equipamentos que permitam uma melhor experiência virtual, é possível cada vez mais encontrar estudos ou pesquisas que envolvam modos de utilizar simuladores virtuais como meio de treinamento nas mais diversas áreas. O uso desses simuladores permite um treinamento sem risco, de custo mais baixo e de fácil reprodução, podendo assim, atingir um maior número de pessoas sem a necessidade de trabalho extra na reprodução e duplicação do treinamento, e em casos de sistemas que não necessitem de dispositivos especiais, sem custo extra.

Para uma melhor experiência, e conseqüentemente melhor aprendizado, é interessante o uso de Realidade Virtual (RV), que pode ser definida como uma “interface avançada do usuário” para acessar aplicações no computador. Esta interface possui a visualização e movimentação em ambientes tridimensionais, em tempo real, e proporciona interação com elementos desse ambiente (TORI e KIRNER, 2006).

O setor elétrico pode usufruir do uso de simuladores, visto que é repleto de operações complexas e perigosas, que proporcionam risco de morte e risco a patrimônio tangível e intangível. É importante, portanto, que os profissionais do setor sejam altamente qualificados, pois não são admitidos erros operacionais.

No relatório de estatística de acidentes do setor elétrico brasileiro de 2012, apresentado na Tabela 1, realizado pelo Comitê de Gestão Empresarial (COGE) é possível ver o número de acidentes ocorridos no setor elétrico nos últimos anos. Em 2012, por exemplo, ocorreram 1234 acidentes em empresas, 696 dos quais envolveram afastamento do funcionário, acarretando, entre custos diretos (remuneração do empregado durante seu afastamento) e indiretos (custo de reparo e reposição de material, custo de assistência ao acidentado e custos complementares – interrupção de fornecimento de energia elétrica, por

exemplo), prejuízos de monta para o Setor de Energia Elétrica, da ordem de centenas de milhões (COGE, 2013).

Como pode ser visto em (COGE, 2013) os acidentes fatais, ao longo dos anos, têm como causas principais: queda, origem elétrica e veículos. Tais causas podem ser evitadas, especialmente as duas primeiras, pois dependem exclusivamente do cumprimento de procedimentos técnicos de trabalho (planejamento da segurança no trabalho, observação das frentes de trabalho, procedimentos de trabalho escritos - o passo a passo, treinamento da força de trabalho, além do compromisso gerencial, etc.), elementos constantes do Sistema de Gestão do Trabalho Seguro (SGTS).

Tabela 1 - Relatório de Estatística de Acidentes do Setor Elétrico Brasileiro – 2012.

Indicadores	Ano						
	2006	2007	2008	2009	2010	2011	2012
1 - Nº de Empregados Próprios (média)	101.105	103.672	101.451	102.766	104.857	108.005	108.133
2 - Horas-homem de Exposição ao Risco	200.219.744	201.981.289	203.945.395	201.104.170	207.109.916	217.351.899	220.225.101
3 - Acidentados Típicos das Empresas							
Acidentados com Afastamento	840	906	851	781	741	753	696
Acidentados sem Afastamento	918	897	901	763	651	595	538
Total	1.758	1.803	1.752	1.544	1.392	1.348	1.234
Consequência Fatal	19	12	15	4	7	18	9
Taxa de Frequência	4,20	4,49	4,17	3,58	3,58	3,46	3,16
Taxa de Gravidade	719	538	568	238	337	581	409
4 - Tempo Computado Total (dias)	144.018	108.756	115.748	47.920	69.853	126.236	90.004
5 - Nº de Empregados das Contratadas (média)	110.871	112.068	126.333	123.704	127.584	137.525	146.314
6 - Acidentados das Contratadas							
Consequência Fatal	74	59	60	63	72	61	58
7 - Acidentados da População							
Consequência Fatal	293	324	331	288	308	309	292

Fonte: (COGE, 2013)

Tendo em mente o que foi mostrado acima, é importante o treinamento eficaz dos operadores da área elétrica. Para isso, esse trabalho procura melhorar a experiência de usuário com o simulador de procedimentos no setor elétrico ATreVEE 3D, apresentado em (SANTOS, 2013), com a adição de Interação Natural (IN), usando como meio de entrada comandos gestuais.

1.2. Objetivos

Este trabalho tem como **objetivo geral** projetar e executar a integração de interação natural em simuladores de procedimentos para o setor elétrico e fazer testes com voluntários do setor.

Para alcançar o objetivo geral, os seguintes objetivos específicos foram buscados:

- Analisar a viabilidade de integração de ferramentas de interação natural em simuladores de procedimentos para o setor elétrico;

- Projetar a adição de interação natural em simuladores de procedimentos para o setor elétrico;
- Executar a integração de interação natural em simuladores de procedimentos para o setor elétrico;
- Fazer um estudo de caso da solução proposta, no simulador ATreVEE 3D;
- Realizar testes com usuários com voluntários do setor elétrico.

1.3. Organização do Trabalho

Este trabalho está configurado em cinco capítulos estruturados da seguinte forma. O Capítulo 1 (Introdução) contextualiza o projeto, descrevendo o problema encontrado e a solução proposta. O Capítulo 2 (Fundamentação Teórica) apresenta conceitos e ferramentas utilizados no desenvolvimento de interfaces. O Capítulo 3 (Simuladores de Treinamento) mostra vantagens proporcionadas pelo uso de simuladores em treinamentos, e apresenta trabalhos desenvolvidos em diferentes áreas, principalmente no treinamento no Setor Elétrico. O Capítulo 4 (ATreVEE 3D) apresenta a primeira versão do ATreVEE 3D, mostrando sua arquitetura e como é a interface com o usuário. O Capítulo 5 (ATreVEE IN) apresenta a nova versão do simulador, descrevendo a interface com o usuário, os cenários que o compõem e diversos aspectos do desenvolvimento. O Capítulo 6 (Testes com Usuários e Análise dos Resultados) descreve os testes realizados com usuários e lista e discute os resultados obtidos. O Capítulo 7 (Considerações Finais) lista algumas dificuldades encontradas ao longo do trabalho, além de mostrar as contribuições geradas e sugerir alguns trabalhos futuros.

2

Fundamentação Teórica

2.1. Realidade Virtual

Realidade Virtual (RV) é definida por (FITZGERALD e RIVA, 2001) como um ambiente tridimensional gráfico sintetizado por computador. Utilizando dispositivos de saída visuais e auditivos, o operador humano pode experimentar o ambiente como se fosse parte do mundo, podendo esse mundo gerado ser um modelo do mundo real, ou um mundo abstrato que não existe num sentido real, mas é entendido pelas pessoas (como a representação de um conjunto de dados) ou pode ser algo totalmente imaginário. Sendo assim RV torna-se uma importante ferramenta em todas as áreas, pois permite representações criadas de cenários (ou dados), reais ou não, de acordo com a necessidade da aplicação desejada. Ela possibilita a criação de uma representação gráfica de uma informação ou situação com a qual é possível interagir.

Pinho (2000) define um Ambiente Virtual como cenário dinâmico armazenado em computador e exibido através de computação gráfica. Ele mostra que o desenvolvimento de uma interface para o ambiente virtual é realizada para efetuar operações de **seleção**, **manipulação** e **navegação**. A operação de **manipulação** seria a tarefa de modificar algum parâmetro do objeto tridimensional, que possa ocasionar alguma alteração às suas propriedades geométricas ou não. **Seleção** seria o processo de definir quais dos objetos do ambiente virtual se deseja operar. E a **navegação** é o processo de mudar a posição do observador do ambiente virtual, permitindo uma nova visão do ambiente ou a exploração de novas partes dele.

O usuário pode perceber o mundo virtual através de uma janela, sendo essa constituída pela tela do monitor ou pela tela de projeção ou ser inserido no mundo virtual, através de capacete (HMD) ou de salas com multiprojeção (cavernas) e dispositivos de interação. A realidade virtual implementada no modo janela é denominada não imersiva, enquanto que a implementação baseada em capacete (HMD) ou salas de multiprojeção e em

outros dispositivos multissensoriais é denominada imersiva (KIRNER e KIRNER, 2011), exemplificadas na Figura 1.

Figura 1 - Realidade virtual não-imersiva e imersiva.



Fonte: (KIRNER e KIRNER, 2011).

Os SRVs estão proporcionando, sem sombra de dúvida, algumas mudanças no processo educacional. O uso de Sistemas de Realidade Virtual (SRVs) como um ambiente de apoio à aprendizagem, onde o usuário possa se movimentar, ouvir, ver e manipular objetos, como no mundo real, geram interessantes oportunidades à disposição dos educadores (YONCK, 2010). Na Tabela 2 são listadas algumas áreas potenciais onde os benefícios dos SRVs educacionais podem ser esperados.

Tabela 2 – Uso potencial dos SRVs.

Uso Potencial	Benefícios comparados com os métodos tradicionais
Simulação de sistemas complexos	Habilidade para observar a operação do sistema a partir de um número de perspectivas aliadas a uma alta qualidade de visualização e interação.
Visualização macroscópica e microscópica	Observação de propriedades de objetos, que são muito grandes ou muito pequenos para serem observados em escala normais.
Simulação em tempos mais rápidos ou mais lentos	Habilidade para controlar a escala de tempo em um evento dinâmico. Esta facilidade pode operar como no avanço ou retrocesso rápido dos gravadores de vídeo modernos.
Altos níveis de interatividade	Os SRVs permitem um grau maior de interatividade que outros sistemas baseados em computador.
Sensação de imersão	Em algumas aplicações, a sensação de escala é extremamente importante.
Flexibilidade e adaptabilidade	Um mesmo SRV pode ser alocado a diversos usos.

Fonte: (FERREIRA, TAROUCO e BECKER, 2003).

2.2. Interação Natural

A Interação Natural trata de criar sistemas que entendem ações naturalmente utilizadas pelas pessoas para se comunicar, enquanto permite aos usuários interagir entre eles e com o ambiente a seu redor. Por usar ações comumente utilizadas na vida real, a Interface

Natural persuade os usuários que eles não estão lidando com o abstrato, mídia digital, e sim com o físico, objetos reais, isso diminui a carga cognitiva, resultando em uma maior atenção ao conteúdo (VALLI, 2007). A Interação Natural ganhou popularidade com o advento dos jogos movidos a gestos a exemplo do Nintendo Wii, cujos controles permitem uma interação diferente dos “antigos” consoles, não só apertando botões, mas apontando, mexendo, girando, oferecendo um mundo de novas interações. A partir da inovação da Nintendo, surgiram variações como o PlayStation Move da Sony que funciona com o Playstation 3, Já Microsoft trouxe com o Kinect ainda mais inovação, uma vez que permite o uso do próprio corpo e seus movimentos para interação (YONCK, 2010).

Para o desenvolvimento de uma interface mais natural, é necessário o estudo de meios de interações que tornem a experiência virtual mais intuitiva e agradável, como pode ser visto no trabalho de treinamento laparoscópico apresentado por Panait *et al.* (2009). Existem diversas pesquisas em diferentes modos de interação como podem ser vistas no trabalho de Turk (2014). A dimensão lúdica está sempre presente na Interação Natural o que é interessante quando queremos representar e reproduzir a realidade. No caso especial de sistemas de treinamento, essa característica representa o poder de captar a atenção dos aprendizes de modo que eles não se preocupem com a fadiga e não fiquem facilmente entediados. Essa é a grande vantagem da implementação da Interação Natural em diversas aplicações.

Existem diversos tipos de interface de interação natural, por exemplo, interface de gestos, de locomoção, hápticas, de voz, cada uma delas busca usar algum sentido humano como forma de interagir mais naturalmente com o usuário. Um ponto importante é que não se pode afirmar que uma delas é mais eficiente que a outra, até mesmo levando em consideração a interface tradicional (mouse e teclado), pois a eficiência de uma interface está diretamente relacionada com o objetivo da aplicação em que a mesma está inserida.

2.2.1. Meios de Interação Natural

Na tentativa de criar uma experiência mais agradável e mais intuitiva para os diversos sistemas computacionais procura-se o desenvolvimento com diferentes meios de interação, buscando aqueles mais adequados para as particularidades de cada sistema. Novas formas de fornecer dados de entrada para o computador são estudadas com o objetivo de aumentar as possibilidades de desenvolvimento, para atender às necessidades dos sistemas e aperfeiçoar a interação. A seguir, têm-se alguns exemplos de meios de interação.

2.2.1.1. Telas Sensíveis ao Toque

Telas Sensíveis ao Toque permitem o usuário a interagir diretamente com o que está sendo exibido. Essa tecnologia é muito utilizada atualmente, principalmente em *smartphones* e *tablets*. Pode-se citar como exemplo de tecnologias para o desenvolvimento com esse modelo de interação, o Android SDK (ANDROID, 2015), iOS SDK (APPLE, 2008) e o Surface SDK (MICROSOFT, 2011).

2.2.1.2. Sistemas Hápticos

Sistemas hápticos usam tecnologia que possui retorno de força permitindo interação com sensação de tato. Esse tipo de interação pode ser realizado com uso de braços mecânicos, exoesqueletos, luvas, varinhas, etc. Um exemplo de dispositivo háptico é o Geomagic® Touch™, apresentado na Figura 2, (GEOMAGIC, 2013a) (antigo *Sensable PHANTOM Omni*) que possui o *toolkit* OpenHaptics (GEOMAGIC, 2013b) para auxiliar com o desenvolvimento de aplicações.

Figura 2 – Geomagic® Touch™.



Fonte: Website 3DS Claytools¹.

2.2.1.3. Reconhecimento de Gestos/ Movimentos

Permite o uso de gestos como modo de interação com o sistema. Existem 3 tipos principais de tecnologias de sensores aplicadas para esse tipo de interação: câmeras *stereo*, câmeras *time-of-flight* (ToF) e câmeras de luz estruturada (CHEN, WEI e FERRYMAN, 2013).

As câmeras *stereo* são inspiradas na visão humana e constroem a estrutura 3D a partir de duas, ou mais, imagens de pontos de vista diferentes. Devido à complexidade dos cálculos para a construção do mapa de profundidade e por serem sensíveis a mudanças de luz, as câmeras *stereo* não são uma boa opção para aplicações no mundo real e em tempo real (CHEN, WEI e FERRYMAN, 2013).

¹ Disponível em: <<http://www.3ds.co.jp/ClayTools/>>. Acesso em: Dezembro de 2013.

As câmeras ToF estimam a distância de um objeto através de pulsos de luz emitidos por uma única câmera, cujo tempo de reflexão no objeto junto com a velocidade da luz mostra a distância. Em 2013, quando foi feito o levantamento dos dispositivos disponíveis, as câmeras ToF possuíam resoluções entre 64X48 e 200X200 pixels e alta taxa de quadros (*frames*) devido à simplicidade do cálculo das distâncias. Suas principais vantagens eram sua velocidade, o que permite o uso em aplicações em tempo real, e a densidade do seu mapa de profundidade. Suas principais desvantagens eram seu alto preço (a câmera SwissRanger™ SR4000, mostrada na Figura 3, custa cerca de 4300 dólares²) e sua resolução baixa (CHEN, WEI e FERRYMAN, 2013).

Figura 3 – Câmera ToF SwissRanger™ SR4000.



Fonte: Anúncio da loja Acroname².

Sensor de luz estruturada (*structured light*) é a classificação na qual se encaixa o Microsoft Kinect³, apresentado na Figura 4, lançado em 2010 com preço acessível, que permitiu o desenvolvimento de diversos trabalhos com a ajuda de bibliotecas como OpenNI e posteriormente com o SDK oficial da Microsoft. O Kinect gera imagem com resolução de 640X480 e possui uma taxa de 30 quadros por segundo.

Figura 4 - Kinect para XBOX.



Fonte: Web site da console Xbox 360³.

O sensor de profundidade do Kinect é composto por duas partes: um projetor e uma câmera VGA de Infravermelho (*IR - Infrared*). O projetor emite um laser pelo ambiente de

² Disponível em: <<http://acroname.com/products/MESA-IMAGING-SR4000>> Acesso em: Junho de 2015.

³ Disponível em: <<http://www.xbox.com/en-US/kinect>> Acesso em: Janeiro de 2014.

visão, que a câmera capta para criar o campo de profundidade, que é a imagem composta pelos pixels que o Kinect recebe de volta medidos por cores variadas, dependendo da distância que se encontram do sistema, como pode ser visto na representação da Figura 5.

O projetor IR emite um padrão em infravermelho que é captado pela câmera IR e então, conhecendo o padrão emitido, é analisada a distorção do padrão na imagem capturada e calculado a distância do ponto analisado (GIL-O-TOPIA, 2010). A partir da distância de cada ponto, é criada uma imagem de profundidade que é exibida com os pixels de cores variadas, que dependem da distância que se encontram do dispositivo (BUNKER, 2011).

Figura 5 - Funcionamento do Kinect.



Fonte: Artigo da PrimeSense Natural Interaction (PRIMESENSE, 2011b).

O software executa um conjunto de filtros sobre essa imagem, para que o Kinect possa detectar o que é o usuário, seguindo um conjunto de especificações e técnicas que definem o que é necessário para que seja um usuário. A partir daí são feitas a identificação das partes do corpo e dos movimentos realizados (BUNKER, 2011).

A vantagem do sensor de luz estruturada estava no seu preço, bem abaixo do preço dos sensores *ToF*. Mas possuía a desvantagem de apresentar buracos na sua imagem de profundidade, pois algumas regiões não podem ser vistas pela câmera e projetor.

2.2.1.4. Reconhecimento de Fala

Proporciona a tradução da fala para texto, permitindo o uso de comandos de voz como modo interação com o sistema. Como exemplos para o desenvolvimento com reconhecimento de fala tem-se a API de Fala oferecida pela Microsoft (*Speech API - SAPI*) (MICROSOFT, 2009), o HTK (*Hidden Markov Model Toolkit*) desenvolvido no Departamento de Engenharia da Universidade de Cambridge (HTK, 1995) e RDC (*Reusable Dialog Components*) desenvolvido pela IBM (HUERTA, LUBENSKY, *et al.*, 2004).

2.2.1.5. Sistemas Controlados pelo Cérebro

Interfaces Cérebro-Computador (*brain-computer interfaces – BCIs*) são dispositivos de comunicação que permitem usuários enviar comandos a um dispositivo de computador usando somente atividade cerebral (VOURVOPOULOS e LIAROKAPIS, 2013).

BCIs podem ser classificados em 3 categorias principais, baseado no meio de gravação dos sinais do cérebro: invasivo, parcialmente-invasivo e não-invasivo. As BCIs invasivas possuem os sinais gravados por eletrodos implantados no cérebro (na massa cinzenta), as parcialmente-invasivas se encontram dentro do crânio, mas não na massa cinzenta, e as não-invasivas utilizam sensores no couro cabeludo sem serem cirurgicamente implantados, sendo o tipo mais utilizado (VOURVOPOULOS e LIAROKAPIS, 2013).

Esse método de interação é uma ótima alternativa para uso em sistemas voltados à pacientes com doenças neurológicas que causem deficiência motora, pois não há a necessidade de nenhum movimento, seja ele para mover algum dispositivo (como o mouse), para realizar algum gesto, ou até mesmo falar.

2.2.1.6. Impulsos Elétricos Musculares

A empresa Thalmic Labs desenvolveu o dispositivo Myo, que pode ser visto na Figura 6, que permite a interação via gestos a partir dos impulsos elétricos dos músculos (THALMIC LABS, 2014).

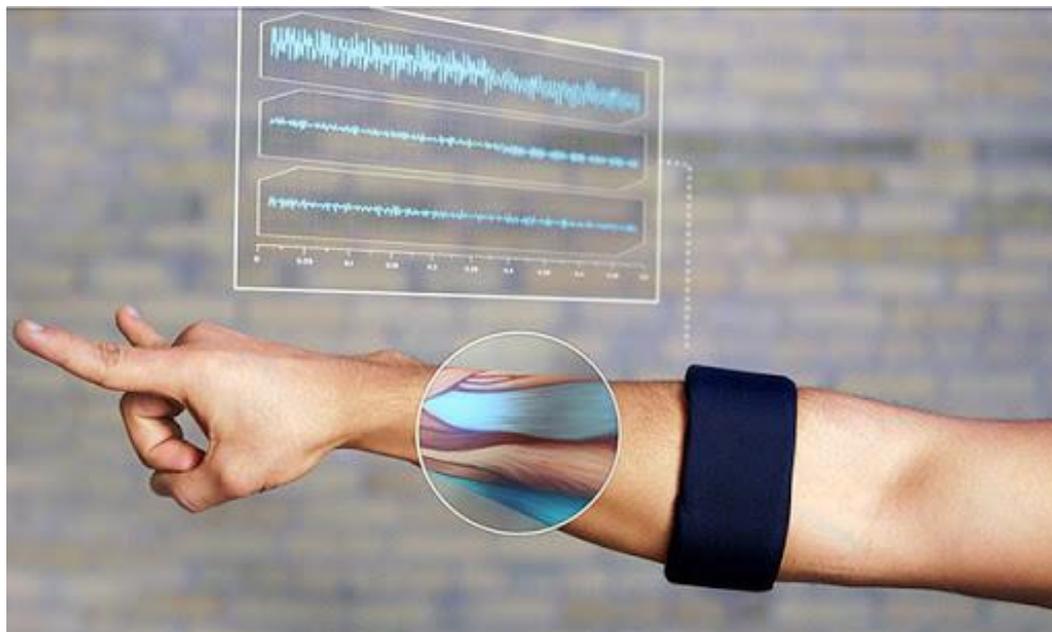
Figura 6 – Myo.



Fonte: Web site do dispositivo Myo da empresa Thalmic Labs⁴.

O dispositivo vem em um formato de bracelete com sensores eletromiográficos (EMG) integrados para captar os sinais elétricos enviados para os músculos do braço identificando as ações realizadas com os dedos, e também o movimento e rotação da mão e antebraço, como representado na Figura 7. Essas informações colhidas são enviadas para qualquer dispositivo digital via *bluetooth*.

Figura 7 - Funcionamento do Myo.



Fonte: Artigo no web site QuaseTudoOnline⁵.

2.3. Controle Leap Motion

O Controle Leap Motion (*Leap Motion Controller*) (LEAP MOTION, 2013), mostrado na Figura 8, é um dispositivo desenvolvido pela empresa Leap Motion em 2012 que permite o reconhecimento e rastreamento de movimentos. O controle é capaz de detectar mãos,

⁴ Disponível em: <<https://www.thalmic.com/en/myo/>>. Acesso em: Abril de 2014.

⁵ Disponível em: <<http://quasetudoonline.com/noticias/conheca-o-myo-bracelete-que-permite-controlar-dispositivos-atraves-do-movimento>>. Acesso em: Abril de 2014.

dedos e ferramentas parecidas com dedo, com alta precisão e alta taxa de quadros rastreados, permitindo mapear suas posições, seus movimentos e gestos.

Figura 8 - Controle Leap Motion.



Fonte: Web site da empresa Leap Motion⁶.

O hardware mostrado na Figura 9 que compõe o controle Leap Motion é composto por 2 câmeras stereo e 3 LEDs infravermelhos. O controle USB do dispositivo lê os dados captados pelos sensores para sua memória local e realiza ajustes na resolução que sejam necessários. Esse dado é então enviado para o software de rastreamento do Leap Motion via USB (COLGAN, 2014).

Quando os dados da imagem são enviados para o computador, depois de compensar pelos objetos de background e a luz ambiental, as imagens são analisadas para reconstruir a representação 3D do que o dispositivo vê. O Algoritmo interpreta os dados 3D e infere a posição de objetos obstruídos, expressando os resultados como uma série de quadros contendo todos os dados do rastreamento. Depois, as bibliotecas organizam os dados em uma estrutura de API orientada a objetos, gerenciam o histórico de quadros e proveem classes e funções auxiliares (COLGAN, 2014).

Figura 9 - Hardware do controle Leap Motion.



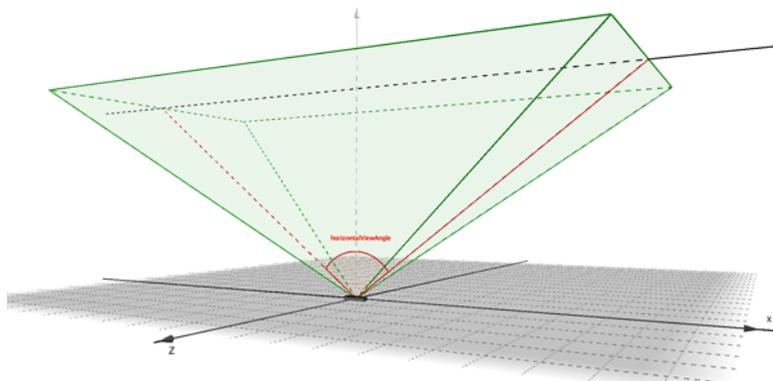
Fonte: Artigo do blog da Leap Motion (COLGAN, 2014).

O controle possui uma área de visualização que pode ser vista como uma pirâmide invertida a partir do dispositivo, que vai de 25 a 600 milímetros verticalmente, como pode ser

⁶ Disponível em: <<https://developer.leapmotion.com/documentation/java/api/Leap.Device.html>>. Acesso em: Março de 2014.

visto na Figura 10, esse limite ocorre devido à propagação da luz do LED pelo espaço, já que fica difícil inferir a posição da mão a partir de certa distância (COLGAN, 2014).

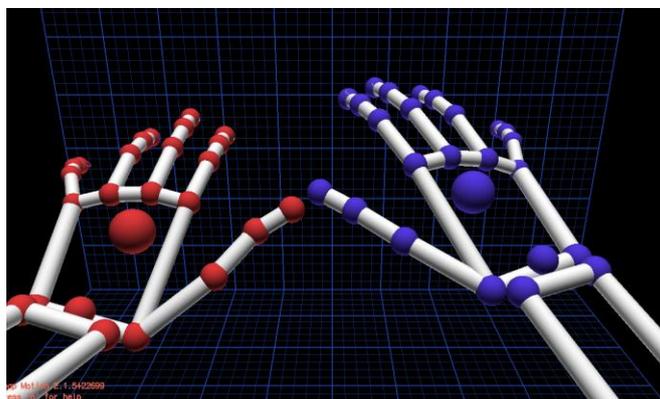
Figura 10 - Campo de visualização do Leap Motion.



Fonte: Documentação do SDK da Leap Motion⁷.

Para o desenvolvimento de aplicações com o dispositivo a empresa Leap Motion disponibiliza um SDK oficial multi-linguagem e multi-plataforma. Na segunda versão do Leap Motion SDK foram feitas diversas melhorias, como a melhora da confiança do rastreamento e a disponibilização do *skeleton*, composto pelos pontos de articulação das mãos rastreadas, como mostrado na Figura 11. O controle Leap Motion foi escolhido para ser usado devido a sua maior precisão em relação ao Microsoft Kinect, e como o cenário é composto de ações com as mãos, o dispositivo Leap Motion se torna mais adequado, uma vez que não há necessidade do rastreamento do corpo todo.

Figura 11 - Skeleton das mãos.



Fonte: PrintScreen de execução de aplicativo de visualização da Leap Motion.

⁷ Disponível em: <<https://developer.leapmotion.com/documentation/java/api/Leap.Device.html>>. Acesso em: Março de 2014.

3

Simuladores de Treinamento

O treinamento por execução no ambiente de trabalho é muito eficiente, principalmente em atividades complexas onde o executor da tarefa possui um grande grau de liberdade, porém é o método de treinamento mais caro, além de ser limitado pela indisponibilidade frequente do contexto do treinamento desejado (PONDER, HERBELIN, *et al.*, 2003). Em busca de solucionar esse problema, diversos sistemas de treinamento virtuais têm sido desenvolvidos nas mais diversas áreas. Este capítulo reúne trabalhos que buscam a melhora de treinamentos e ensino com o uso de simuladores, esses trabalhos são aqui divididos em três seções: treinamento com realidade virtual treinamento no setor elétrico e treinamento com interação natural. E posteriormente, comparados quanto à área de aplicação, o tipo do simulador de treinamento e que meios de interação utilizam.

Um exemplo dos estudos em busca de melhoria no treinamento pode ser visto no trabalho de El-Khalili e Brodlie (2000), no qual há um foco no uso da *Web* para aprimorar o treinamento e ensino na área da saúde, julgando-a ser um ótimo ambiente para treinamento, uma vez que oferece uma grande quantidade de informação e suporte para o aprendizado em grupo à distância. Além de serem, na maioria das vezes, mais baratos e fáceis de reproduzir do que os treinamentos convencionais, os sistemas de treinamento virtuais apresentam diversos outros potenciais benefícios, como congregado por Mantovani (2001):

- *Aprendizado experiencial e ativo;*
- *Visualização e reificação:* ambientes virtuais proporcionam um método alternativo de apresentar informações, além de possibilitar transformar o imperceptível em perceptível (como por exemplo, representações gráficas de informações e ideias);
- *Aprendizado em contextos impossíveis ou difíceis de experimentar na vida real;*

- *Melhoria da motivação*: pois permite para atividades monótonas, o desenvolvimento de treinamentos mais interessantes do que as ações reais, como por exemplo, o uso de jogos;
- *Promoção da colaboração*: sistemas podem ser usados para encorajar a colaboração com treinamento de grupos realizando tarefas em conjunto em ambiente virtual comum;
- *Adaptabilidade*: pois os sistemas podem ser adaptáveis às características dos usuários;
- *Avaliação*: as sessões em um ambiente virtual podem ser facilmente monitoradas, portanto facilitando a avaliação dos usuários.

É possível classificar esses sistemas de treinamento virtuais em duas categorias como apresentado por Ponder *et al.* (2003): treinamento de habilidades e treinamento de situação (ou treinamento de decisão). O sistema para treinamento de manutenção de redes elétricas, no qual o trabalho proposto será realizado, seria classificado na primeira categoria (treinamento de habilidade) que trata dos sistemas que ensinam uma habilidade física (ou motora) ou procedimento, como cirurgia, operação de máquinas e simulação de voo. O treinamento de situação procura ajudar a treinar as habilidades psicológicas do usuário necessárias para situações reais, através de sistemas que apresentam uma experiência sintética semirreal.

3.1. Treinamento com Realidade Virtual

O uso de RV na área de treinamento virtual é uma ferramenta promissora com grande potencial para aprimorar a experiência de aprendizado, pois ambientes virtuais podem gerar um contexto educacional rico, interativo e atraente e proporcionam um meio de aprender na prática, como dito por Mantovani (2001). Ambientes virtuais também possibilitam uma melhor representação da situação real tornando a experiência mais proveitosa, além de permitir outras ações não possíveis no mundo real, como a visualização de estruturas internas de equipamentos ou até mesmo, no caso da medicina, de órgãos. Existem diversos trabalhos que utilizam RV com o intuito de possibilitar e melhorar o treinamento nas mais diversas áreas, a seguir serão mostrados alguns exemplos.

Vora *et al.* (2002) apresenta o desenvolvimento e teste de um simulador para o treinamento de inspeções visuais de aeronaves, formado por uma tarefa de inspeção em um compartimento de uma aeronave de carga. O texto mostra a importância de um bom

treinamento dos inspetores, sendo uma tarefa que não é possível ser aprendida somente com inspeções reais, principalmente pela demora no retorno da inspeção realizada. No trabalho são descritos testes com usuários, que mostram resultados com alto grau de presença (imersão), neles se observa preferência dos usuários em relação a um programa de treinamento que não utilizasse RV.

Outro trabalho na área da indústria aeroespacial é apresentado por Wasfy, Wasfy e Noor (2004), no qual se tem o desenvolvimento de um agente inteligente que juntamente com um ambiente de realidade virtual procuram auxiliar o treinamento da operação de um túnel de vento da NASA. O simulador possui a sala de controle, com painéis de controle funcionais, o túnel de vento e um agente inteligente para guiar os operadores.

O trabalho apresentado por Ponder *et al.* (2003) trata-se do uso de simuladores de treinamento de decisão, em que é importante o enredo do cenário criado, para que haja momentos para decisões serem tomadas que afetem o decorrer cenário. Para uma melhor experiência o simulador de propósito geral apresentado utiliza um rastreador magnético na cabeça do usuário, para ajudar a navegação pelo ambiente, e uma tela (com projeção por trás) com imagem estereoscópica (juntamente com o uso de *shutter glasses*). Ele também possui um assistente virtual controlado (indiretamente) por comando de voz. As ações do assistente e suas respostas e falas são controlados por um supervisor, que pode acionar eventos de cena e as ações e falas do assistente, portanto quando o treinando dá um comando, o supervisor decide o que será feito pelo assistente.

O caso de uso apresentado neste trabalho foi um cenário para ajudar o treinamento de primeiros socorros de paramédicos (incluindo cidadãos voluntários). O treinamento é composto por 3 aspectos: conhecimento, habilidade e treinamento com execução no trabalho. O terceiro aspecto tem sido realizado através do socorro de pessoas, em que não há espaço para erros. Nessa direção, o simulador procura proporcionar esse treinamento por execução de modo que estejam associados a ele o estresse e as consequências dos erros em diferentes situações de emergência, em que o treinando tem que avaliar a situação e decidir quais procedimentos devem ser realizados para tal situação.

Outro trabalho para o treinamento de situações de emergência foi desenvolvido por Hanford, Rogers e Cross (2011), que procura realizar o treinamento de agentes de trânsito para situações de evacuação. No cenário de evacuação os agentes de trânsito podem fazer um número de decisões que afetam o tráfego, incluindo informar os motoristas através de

mensagens. O treinando ainda pode usar bloqueios de rua, polícia de trânsito e controle sobre os semáforos. O simulador utiliza um modelo de comportamento baseado em agente para controle das ações dos veículos durante a evacuação. Cada motorista é representado por um agente que possui um modelo comportamental individual. Esses agentes são proativos para alcançarem seus objetivos e reagem a mudanças no ambiente.

3.2. Treinamento no Setor Elétrico

É possível encontrar alguns bons trabalhos que buscam a utilização de sistemas de treinamentos na área do setor, a seguir têm-se exemplos desses trabalhos correlatos.

Cardoso *et al.* (2013) mostra o trabalho desenvolvido pela Faculdade de Engenharia Elétrica da Universidade Federal de Uberlândia (UFU) em parceria com a empresa de energia elétrica de Minas Gerais CEMIG, o sistema VRCEMIG. Esse sistema permite o controle de subestações pelo ambiente virtual em tempo real. As mudanças no sistema afetam a subestação real e a mudança nos componentes da subestação atualiza o sistema em tempo real. Com isso procura-se proporcionar um ambiente que permite controle das diversas subestações da empresa, como um sistema que permite o treinamento de operadores em ambiente seguro e de baixo risco. O trabalho apresenta também pesquisas realizadas pelo Grupo de Realidade Virtual e Aumentada da UFU com sensores EMG ou reconhecimento de gestos juntamente com realidade aumentada.

O trabalho apresentado por Belloc (2011) propõe um arcabouço para auxiliar o desenvolvimento de simuladores de procedimento com RV e para validar tal proposta são utilizados dois casos de uso na área elétrica. O primeiro é um simulador para a troca de cruzeta em linha viva, enquanto que o segundo envolve a prática da montagem e desmontagem de uma Unidade Geradora Hidrelétrica.

Com o objetivo de aprimorar o processo de capacitação das equipes de manutenção, o primeiro sistema desenvolvido trata-se da execução da troca da cruzeta, que é usada para sustentar todas as fases da rede primária de um poste em linha viva, procedimento esse demorado e de alto risco, por isso a importância de um bom treinamento. A aplicação utiliza como meio de interação o mouse e em adição aos modelos tridimensionais, disponibiliza vídeos para a demonstração de ações mais complexas, caso o usuário deseje usá-los. As ações são escolhidas através de um menu de contexto, com a lista de opções disponíveis, que é apresentado ao selecionar um objeto interativo no cenário.

O segundo sistema desenvolvido, realizado com o objetivo de enriquecer o material utilizado no treinamento, trata-se de procedimentos de montagem e desmontagem de uma Unidade Geradora, através de execução meramente ilustrativa (com a manipulação somente das principais peças da UG). Ao contrário do primeiro estudo de caso, esse cenário não tem enfoque no treinamento de atividades de risco, mas na familiarização e na capacitação do trabalho com máquinas e peças que não estão facilmente disponíveis para o treinamento. Os procedimentos disponíveis nesse cenário quando realizados realmente envolvem uma equipe com vários integrantes e leva semanas para sua conclusão.

Barata (2010) apresenta uma solução de software para a criação de simulações em RV de processos operacionais industriais (um sistema autoral), que permite a criação de simulações sem conhecimento de programação. Para validar o sistema foi desenvolvido um estudo de caso na área do setor elétrico, onde é simulada a partida de uma Unidade Hidrogeradora (UHG). O processo de partida na UHG pode ser monitorado (na Usina Hidrelétrica de Tucuruí) através de um computador, coletando e enviando comandos para a UHG através de sensores e relés instalados por todos os equipamentos da unidade. O processo de partida apresentado não é facilmente observado fora dos monitores de computador, por isso a intenção desse estudo de caso é apresentar todas as alterações ocorridas nos equipamentos da UHG mostrando os próprios equipamentos (o que seria impossível de ser feito na UHG real). Ao mesmo tempo em que o simulador permite o processo ser visto de uma perspectiva única ele também apresenta o processo pela visão do operador, que controla o processo de partida através de computadores.

O Simulador apresentado por Barata permite o usuário realizar as tarefas dos diversos operadores envolvidos no processo de partida real utilizando do mouse e do teclado para interagir com os componentes da cena. O sistema é composto por 82 passos incluindo diversos passos de pré-condições como a abertura da válvula principal do sistema de regulação que é feita manualmente (no processo real) por um operador funcionário da usina. O simulador também apresenta a interface do computador de monitoração, onde o usuário pode utilizar a interface igual a real. Apresentando animações do que ocorre nos equipamentos no decorrer do processo.

Pamplona Jr., Ribeiro Filho e Moreira (2010) apresentam mais um caso de uso do sistema de autoria mostrado por Barata (2010). Dessa vez, o cenário é a manutenção da válvula atuadora do regulador de velocidade do pré-distribuidor da turbina. A válvula atuadora é um componente que transforma um sinal elétrico de pequena potência, emitido

pelo controlador de velocidade em um sinal hidráulico capaz de movimentar a válvula de controle do servomotor principal responsável pela movimentação das pás do rotor e do pré-distribuidor.

Sousa *et al.* (2008) apresenta um ambiente de treinamento de manutenção em uma Unidade Hidrelétrica de Energia (UHE). O trabalho mostra dois módulos controlados via mouse e teclado, um módulo de montagem e um de manutenção. O módulo de montagem permite o treinando executar virtualmente os procedimentos de montagem do concreto e de peças da UHE. O segundo módulo (de manutenção) permite o treinamento de procedimentos de manutenção, realizando operações de desmontagem e montagem de peças e equipamentos da UHE, baseado nos manuais de operação e manutenção da turbina, fornecidos pela ELETRONORTE.

Arendarski, Termath e Mecking (2008) apresentam uma plataforma, desenvolvida no Instituto Fraunhofer, de visualização interativa para desenvolvimento e treinamento virtual. Sistema esse usado para criar cenários animados 3D para diversas funções e áreas, em particular, é mostrado um cenário virtual que apresenta um transformador de energia de voltagem extra alta. O cenário descrito trata da preparação, desmontagem e desconexão de um transformador para que ele seja transportado. O simulador inclui passos preparatórios, como isolar a área de trabalho e preparar os equipamentos que serão utilizados, usando o mouse para selecionar os objetos da cena e juntamente com o teclado manipulá-los.

Los Arcos *et al.* (1999) apresentam um sistema para o treinamento de operadores do setor elétrico constituído de dois sistemas independentes: o Simulador SGI (Sistema Gráfico Interativo), utilizado em treinamento e reciclagem dos processos dos operadores em ações nas subestações onde não há automação ou instalações para controle remoto; e o Assistente, utilizado para auxiliar o usuário no processo de treinamento, provendo assistência quando o usuário erra ou quando requisitado.

Garant *et al.* (1995) apresentam o ESOPE-VR, um sistema de RV utilizado para o treinamento de operários que trabalham em comutação de energia elétrica ou estações de distribuição, permitindo os operários a praticarem rotinas que teriam que realizar no curso do seu trabalho. O maior foco da interação manual no simulador está nos painéis de controle nos quais os equipamentos podem ser operados, para interagir com os botões e alavancas do painel é utilizado um modelo de mão 3D controlado por um dispositivo de 6 graus de liberdade (*6-DoF: 6-Degree-of-Freedom*, que permite o movimento nos eixos x, y, z e a

rotação em cada um dos eixos). O sistema ainda permite o uso de comandos de voz para controlar a direção de movimento e disponibiliza como retorno imagem tridimensional (estereoscópica) e avisos de áudio.

O trabalho apresentado por Buriol (2011) mostra uma plataforma desenvolvida na UFPR (Universidade Federal do Paraná) de treinamento no setor elétrico, projeto esse desenvolvido com apoio da Copel, Concessionária de energia com sede em Curitiba. O sistema possibilita o treinamento da troca de cruzeta de postes elétricos em linha viva, para isso o sistema utiliza dois Wii Remotes (Controle do console Nintendo Wii, que possui sensor infravermelho), duas TVs estereoscópicas e óculos para visualização 3D, LEDs infravermelhos para referências, e dispõe também do uso de uma cesta isolada real, dentro da qual o eletricitista permanece para a manutenção das linhas de transmissões reais.

Os Wii Remotes são utilizados como apontador e para realizar o rastreamento de cabeça. Para manipular interagir com a cena, o usuário utiliza um Wii Remote para apontar para a posição que deseja, além de poder rotacionar e mover a câmera para mudança da área de visualização da cena, isso é feito utilizando LEDs infravermelhos presos no friso do televisor, que são captadas pela câmera infravermelha do Wii Remote e partir da posição do controle em relação às LEDs são calculadas as coordenadas do cursor no cenário. Para o rastreamento de cabeça um par de LEDs são colocados nos óculos para visualização 3D, e um Wii Remote é posicionado em cima das TVs com a câmera infravermelha voltada para o usuário, com as coordenadas dos dois LEDs o sistema rastreia a posição e rotação da cabeça e utiliza esses dados para movimentar a câmera para reproduzir uma visualização que respondam a movimentos feitos pelo usuário.

O simulador permite com o uso do controle a manipulação dos objetos de cena, apontando o objeto desejado e pressionando um botão do controle para “pegá-lo”. O treinamento é realizado com o treinando dentro da cesta isoladora e utilizando os EPIs (Equipamentos de Proteção Individual) para reproduzir as condições que o usuário estaria na situação real.

No trabalho aqui proposto será utilizado como caso de uso o trabalho apresentado por Santos (2013), um simulador de treinamento no setor elétrico, ATreVEE 3D, para troca de isoladores de postes elétricos em linha viva, que será mais bem discutido posteriormente no Capítulo 4. Propõe-se o estudo de interface natural para sua aplicação em sistemas de

treinamento do setor elétrico, para isso será incrementado o trabalho apresentado por Santos em busca de uma melhor interação a fim de aperfeiçoar a experiência.

3.3. Treinamento com Interação Natural

O uso de tecnologias de Interação Natural está cada vez mais presente em trabalhos para treinamento de procedimentos em busca de um sistema que simule a operação real com mais fidelidade, tornando o treinamento mais eficiente e eficaz.

O uso de IN para o treinamento com simuladores é presente nas mais variadas áreas, como na área de manutenção aeroespacial (ABATE, GUIDA, *et al.*, 2009) e até mesmo na área de esporte (MILES, POP, *et al.*, 2012). Abate *et al.* (2009) utilizam tecnologias hápticas no simulador de manutenção de aviões para uma experiência mais rica e Miles *et al.* (2012) fazem uma revisão de simuladores de esportes que muitas vezes utilizam tecnologias personalizadas para melhor se aproximar das ações reais, além de mencionar o trabalho que está desenvolvendo, um simulador para treinamento de rúgbi.

Os dispositivos hápticos são amplamente utilizados em estudos que procuram proporcionar treinamento, pois proporcionam um tipo diferente de retorno, melhorando a percepção que o usuário tem do ambiente, fazendo com que ele não só possa ver e ouvir, mas também sentir. É possível encontrar diversos trabalhos na área de treinamento de procedimentos médicos, como é o caso dos simuladores para treinamento de procedimentos laparoscópicos avaliados no trabalho de Panait *et al.* (2009), no qual foi concluído que em tarefas avançadas o uso de retorno háptico possibilita melhor treinamento, mostrando o potencial que o uso de Interação Natural tem de aprimorar sistemas de treinamento. Outro trabalho que vale ser destacado é o trabalho desenvolvido na UFPB (Universidade Federal da Paraíba) em conjunto com outras universidades, o framework CyberMed incrementado por Cunha *et al.* (2012), que auxilia o desenvolvimento de aplicações para treinamento médico utilizando sistemas hápticos. Alguns outros trabalhos sobre treinamento médico podem ser vistos no trabalho de Gosselin *et al.* (2013) para cirurgias buco-maxilo-faciais e no de Robison, Liu e Apuzzo (2011) com um *review* na área cirúrgica neurológica.

O uso do Kinect também pode ser encontrado para auxiliar o treinamento. Semeraro *et al.* (2013) mostra um sistema criado para auxiliar no treinamento de ressuscitação cardiopulmonar. O treinamento utiliza um manequim, em que são realizadas as compressões, e uma câmera Kinect para captar a imagem do usuário para tratamento. Com as imagens

captadas pelo Kinect do movimento sendo realizado, são analisadas a velocidade e a profundidade das compressões realizadas e indicadas se devem ser aumentadas, diminuídas, ou se estão corretas.

3.4. Comparativo

Na Tabela 3 é apresentado um comparativo dos trabalhos mostrados anteriormente, classificando-os pelo tipo de treinamento e área de atuação, e definindo quais dispositivos são utilizados nos sistemas. Pela tabela é possível observar que apenas 3 trabalhos do Setor Elétrico utilizam dispositivos diferente dos mais usuais (mouse e teclado). Isso se torna uma limitação para a simulação, pois se tratam de dispositivos que fornecem dados de entrada bidimensionais enquanto se tenta interagir com um ambiente tridimensional. Em busca de sanar essa limitação, trabalhos como os de Cardoso *et al.* (2013) , Buriol (2011) e Garant *et al.* (1995), usam diferentes dispositivos para usufruir de uma interação mais rica. Os trabalhos de Buriol (2011) e de Garant *et al.* (1995) merecem destaque, pois usam dispositivos que fornecem dados tridimensionais para serem usados na interação, mas são limitados na navegação e interação com objetos.

No trabalho de Buriol, o movimento da cabeça é usado para mudar a direção da câmera, simulando o movimento real para ter-se um novo campo de visão. Mas para a movimentação para frente e para trás no cenário é necessário o pressionamento de botão juntamente com o movimento do controle. De forma semelhante, é realizada a interação com os objetos, para arrastá-los, é necessário um botão estar selecionado para segurá-los. Garant *et al.* apresentam uma solução que usa uma representação com modelo 3D das mão do usuário dentro da cena, que é controlada com o mouse 6-DoF, e usando a colisão do modelo com os botões e alavancas do painel de controle no cenário para ativá-los. Apesar do controle da mão ser realizada pela reprodução dos movimentos realizados, para a navegação do ambiente foram apresentadas duas alternativas usadas: o uso dos botões no mouse 6-DoF e comandos de voz, sendo esses, comandos que não são uma representação das ações necessárias reais.

No trabalho aqui desenvolvido procura-se usar um dispositivo de rastreamento para reconhecer gestos e utilizá-los para interagir e navegar no ambiente, utilizando-se de gestos que se assemelhem com os realizados nas operações reais, proporcionando assim uma simulação mais próxima do procedimento real.

Tabela 3 - Comparativo de trabalhos.

Trabalho	Tecnologia	Dispositivos	Área	Classificação
(EL-KHALILI e BRODLIE, 2000)	Internet	Mouse/ Teclado	Medicina	----
(WASYF, WASFY e NOOR, 2004)	RV/ IA/ Interface de Linguagem Natural/ Reconhecimento de Fala/ Estereoscopia Imersiva/ Rastreo de Usuário e de Dispositivo	Mouse/ Teclado/ Display Estereoscópico Imersivo (Quarto com Displays como Paredes) / Shutter Glasses/ Alto-Falantes/ Dispositivos de Rastreo/ Varinha Rastreada/ Microfone/ Joystick	Aeroespacial	Treinamento de Habilidade
(VORA, NAIR, et al., 2002)	RV/ Rastreo de Olhos e Dispositivo	Mouse/ <i>Head Mounted Display</i> (HMD) com Rastreador de Olhos/ Sensor <i>Flock of Bird</i> / Mouse 6-DOF	Aeroespacial	Treinamento de Habilidade
(PONDER, HERBELIN, et al., 2003)	RV/ Estereoscopia/ Rastreo da Cabeça	Mouse e Teclado (Supervisor) / Projetor/ Shutter Glasses/ Rastreador Magnético/ Microfone	Uso Geral/ Emergência (Primeiros Socorros) *	Treinamento de Decisão
(HANDFORD, ROGERS e CROSS, 2011)	RV	Mouse/ Teclado	Emergência (Evacuação no Trânsito)	Treinamento de Decisão
(CARDOSO, LAMOUNIER JR., et al., 2013)	RV/ Estereoscopia	Mouse/ Teclado/ <i>Joystick</i> / <i>Gamepad</i> / Óculos	Setor Elétrico	Treinamento de Habilidade
(BURIOL, 2011)	RV/ Estereoscopia/ Rastreo de Cabeça/ <i>Ray Casting</i>	Wii Remotes/ TVs Estereoscópicas	Setor Elétrico	Treinamento de Habilidade
(BELLOC, 2011)	RV	Mouse	Setor Elétrico	Treinamento de Habilidade
(BARATA, 2010)	RV	Mouse/ Teclado	Uso Geral/ Setor Elétrico*	Treinamento de Habilidade
(PAMPLONA JR., RIBEIRO FILHO e MOREIRA, 2010)	RV	Mouse/ Teclado	Setor Elétrico	Treinamento de Habilidade
(SOUSA, PAMPLONA JR., et al., 2008)	RV	Mouse/ Teclado	Setor Elétrico	Treinamento de Habilidade
(ARENDARSKI, TERMATH e MECKING, 2008)	RV	Mouse/ Teclado	Uso Geral/ Setor Elétrico*	Treinamento de Habilidade
(LOS ARCOS,	RV	Mouse	Setor Elétrico	Treinamento de

VICEDO, et al., 1999)				Habilidade
(GARANT, DAIGLE, et al., 1995)	RV/ Estereoscopia/ Reconhecimento de Fala	6-DoF Mouse/ Shutter Glasses/ Microfone	Setor Elétrico	Treinamento de Habilidade
(ABATE, GUIDA, et al., 2009)	RV/ Retorno Háptico	HMD/ CyberForce/ CyberGrasp/ CyberGlove	Aeroespacial	Treinamento de Habilidade
(PANAIT, AKKARY, et al., 2009)	RV/ Retorno Háptico	Mouse/ Teclado/ Dispositivo Háptico	Medicina	Treinamento de Habilidade
(CUNHA, XIAB, et al., 2012)	RV/ Retorno Háptico	Dispositivo Háptico	Medicina	Treinamento de Habilidade
(GOSSELIN, BOUCHIGNY, et al., 2013)	RV/ Retorno Háptico/ Estereoscopia	Dispositivos Hápticos Personalizados	Medicina	Treinamento de Habilidade
(SEMERARO, FRISOLI, et al., 2013)	Reconhecimento de Movimento	Microsoft Kinect/ Manequim RCP	Emergência (Primeiros Socorros)	Treinamento de Habilidade
Trabalho Proposto	RV/ Rastreo de Mãos/ Reconhecimento de Gestos	Mouse/ Teclado/ Controle Leap Motion	Setor Elétrico	Treinamento de Habilidade

**Área de aplicação no trabalho*

Capítulo

4

ATreVEE 3D

Como estudo de caso será utilizado o simulador de procedimentos no setor elétrico ATreVEE 3D (SANTOS, 2013). O simulador pretende auxiliar o treinamento de profissionais do setor elétrico, tendo em vista o alto risco das atividades por eles realizadas. Esses profissionais precisam ser altamente qualificados, e essa ferramenta propõe ajudar em tal qualificação, permitindo também a realização da simulação a qualquer momento para que o operador possa rever os conceitos e instruções para a realização do procedimento quando necessário. Com isso visa-se ajudar os profissionais a trabalharem com mais segurança além de diminuir a ocorrência de problemas na rede elétrica, o que pode afetar diversas pessoas.

O sistema elétrico brasileiro é um sistema hidrotérmico de grande porte, com predominância de usinas de hidroelétricas e longas linhas de transmissão (SANTOS, 2013). Durante o longo percurso da energia elétrica entre as usinas e os centros consumidores nas redes de distribuição existem elementos importantes das redes de transmissão que são os isoladores de vidro ou porcelana, que sustentam os cabos e impedem descargas elétricas durante o trajeto.

Como pode ser visto em Ferreira (2011), as falhas em isolamentos podem causar, desde interferências em sinais de rádio ou televisão, até curtos-circuitos que podem retirar a linha ou subestação de operação. A interrupção temporária de suprimento acarreta prejuízos às unidades consumidoras e à concessionária de energia. Para a concessionária, falhas em isolamentos podem resultar em danos materiais próprios e de terceiros, custos de manutenção, perda de faturamento e, em muitos casos, multas impostas por parte das agências reguladoras.

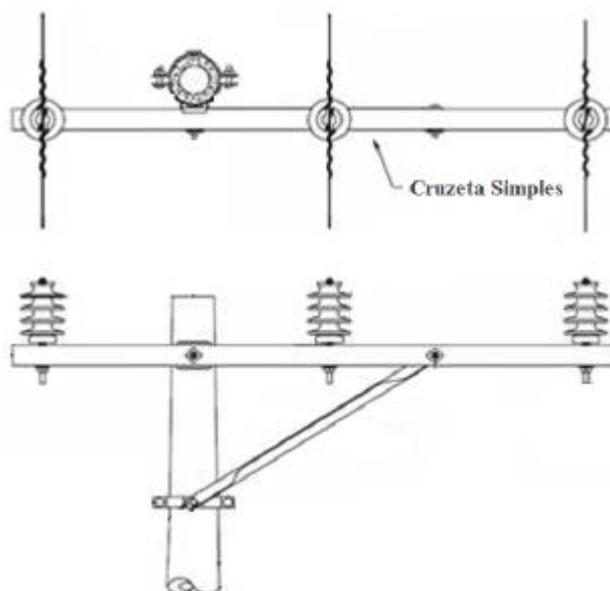
O simulador ATreVEE 3D busca evitar esses problemas com o treinamento do procedimento de substituição de isoladores, mostrados na Figura 12, em linha ativa e desativada em postes trifásicos com tensão de 13,8 kV. Na Figura 13 é ilustrado o poste comum que é utilizado para sustentar todas as fases da rede primária.

Figura 12 - Isoladores usados e, linhas de 13,8 KV.



Fonte: (SANTOS, 2013)

Figura 13 - Poste com Isolador.



Fonte: (BELLOC, 2011)

Na Tabela 4 são apresentados os passos necessários para a troca de um dos isoladores em linha ativa, que devem ser realizados no treinamento. Para a troca do primeiro isolador, se deve colocar inicialmente quatro canaletas protetoras nos dois condutores próximos do isolador, um em cada lado dos dois isoladores que os sustentam, e a cobertura isolante na cruzeta entre os dois condutores. Após serem colocados os protetores, o operador pode soltar o fio de cobre que fixa o condutor ao isolador para depois mover o condutor, tirando-o do isolador e colocando-o na cruzeta sobre a cobertura isolante. Ao tirar o condutor do isolador, o operador retira o fio que está envolvido no isolador, e retira o isolador antigo do poste, trocando-o por um novo. Colocado o novo isolador, o operador deve prender o fio no isolador novo e depois mover o condutor para o novo isolador prendendo-o, nos dois lados, com o fio

de cobre. Para o primeiro isolador, não há a necessidade de remover os protetores, pois esses serão usados para a troca do segundo isolador.

Tabela 4 - Passos para troca do primeiro isolador em linha ativa.

Numeração	Procedimentos
1	Colocar a canaleta protetora 1 no primeiro condutor (cabo de alta tensão).
2	Colocar a canaleta protetora 2 no segundo condutor.
3	Colocar a canaleta protetora 3 no primeiro condutor.
4	Colocar a canaleta protetora 4 no segundo condutor.
5	Colocar a cobertura isolante entre os isoladores.
6	Retirar o lado direito do fio de cobre que prende o condutor ao isolador.
7	Retirar o lado esquerdo do fio de cobre que prende o condutor ao isolador.
8	Mover cabo de alta tensão para cima da cobertura isolante.
9	Retirar fio de cobre do isolador.
10	Retirar isolador antigo.
11	Colocar novo isolador.
12	Colocar novo fio de cobre para fixar o condutor ao isolador.
13	Colocar o cabo de alta tensão em cima do isolador.
14	Fixar o lado direito do fio de cobre no condutor.
15	Fixar o lado esquerdo do fio de cobre no condutor.
16	Encerrar procedimento.

Fonte: Passos descritos no trabalho de Santos (2013).

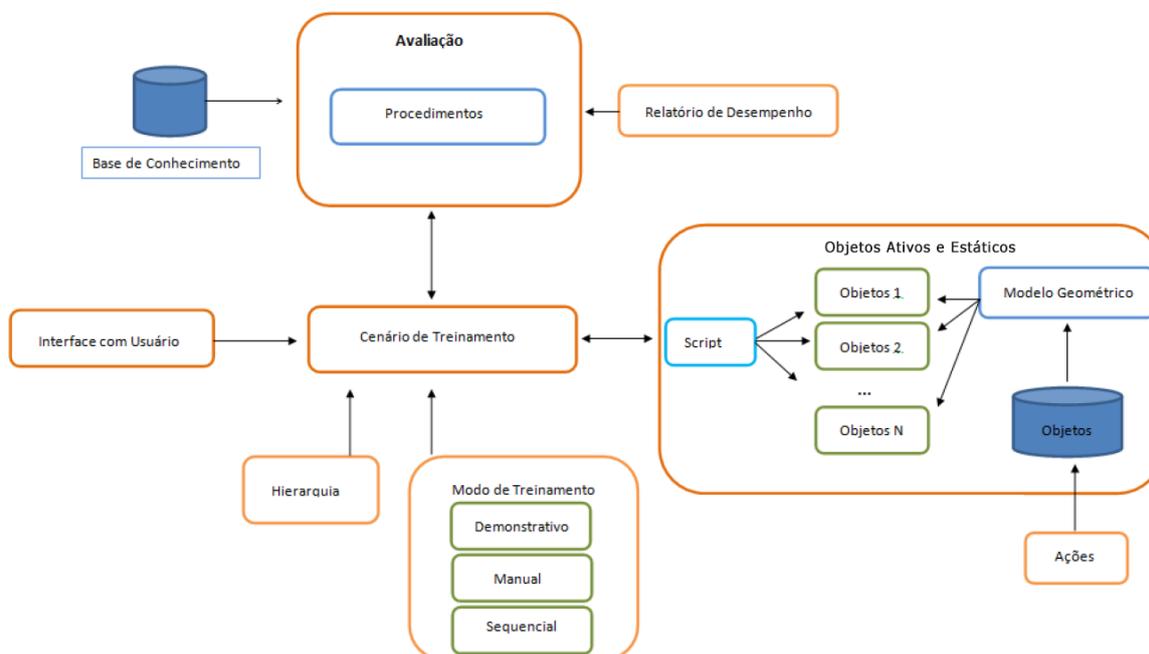
4.1. Esquema Proposto para Simuladores de Procedimento

O trabalho de (SANTOS, 2013) também apresentou um esquema para ser utilizado durante o processo de desenvolvimento de simuladores para o setor elétrico. Como pode ser visto na Figura 14, o esquema apresentado é composto por 4 componentes principais: o cenário de treinamento, a avaliação dos procedimentos, um gerenciador de todos os objetos interativos e estáticos presentes no cenário e a interface com o usuário.

A *avaliação dos procedimentos* pode ser definida como a identificação e mensuração das ações que os operadores realizaram em cenário de operação durante determinado período. A avaliação de desempenho tem como objetivo diagnosticar e analisar o desempenho individual e em grupo dos operadores, promovendo o crescimento pessoal e profissional, bem como melhor desempenho de absorção do conteúdo imposto. A avaliação usa a *base de*

conhecimento, no processo de avaliação de cada usuário, que representa o repositório de informações e referência que os operadores possuem antes de iniciar a utilização do simulador, como o conhecimento dos acessórios necessários para a operação.

Figura 14 - Modelo proposto para desenvolvimento de simuladores de procedimentos.



Fonte: (SANTOS, 2013)

O módulo *objetos interativos e estáticos* contém os *modelos geométricos* e *scripts* que regulam as regras de funcionamento dos objetos interativos. Os *scripts* acionam as diferentes movimentações (ações) que estão relacionados diretamente aos objetos de cada cenário. As ações são os mecanismos que fazem todo o simulador trabalhar, estas ações são organizadas em listas que são criadas para todos os objetos ativos da cena. O *script* torna essas ações em rotinas que quando executadas, realiza a reação do objeto. Quando o operador aciona a rotina referente ao objeto desejado, uma sequência de animações são executadas, por meio delas os objetos selecionados e, quando necessário, outros objetos da cena são modificados de acordo com os resultados da ação no procedimento, podendo modificar aspectos como, posição, textura, cor, etc.

A *interface com o usuário* tem como objetivo tornar a interação do operador o mais simples possível. Assim, um bom projeto de interface com o operador facilita a conclusão da tarefa manualmente sem chamar atenção desnecessária para si, equilibrando a funcionalidade técnica e elementos visuais.

4.1.1. Cenário de Treinamento

O *cenário de treinamento* estabelece a interface com a aplicação. Através desta interface, a aplicação pode controlar todo o processo de treinamento do operador, o avaliando de acordo com sua evolução, interagindo com todos os objetos interativos do cenário e todos os procedimentos disponíveis para a ação corrente. Esse é o elemento principal do simulador, ele representa a construção do sistema e das cenas com seus objetos e regras, através do qual o operador pode consultar e manipular objetos interativos presentes no ambiente e escolher os procedimentos que serão praticados.

4.1.1.1. Modos de Treinamento

O simulador apresenta três distintos modos de treinamento, escolhidos pelo operador antes de começar qualquer procedimento, que são eles: Modo Manual, Modo Sequencial e Modo Automático. O Modo Manual e Sequencial tratam-se da execução do procedimento pelo usuário, no primeiro, realizando apenas o procedimento principal, e no segundo, realizando os passos do operador desde a chegada da empresa, incluindo a entrada na empresa e a escolha dos equipamentos necessários. Já o Modo Automático é composto pela execução automática dos passos do procedimento, para que o operador possa assistir e aprender ou revisar os passos do procedimento antes da execução.

4.1.1.2. Hierarquia de Operações

A execução de ações pelo o operador durante o procedimento é controlada por uma *hierarquia de operações*, que representa os aspectos procedurais do simulador tais como, sequência dos processos, ocorrência/ordem de decisões ou repetições de operações. Apesar das diversas ações que formam os procedimentos, apenas uma pode ser executada por vez, como pode ser visto na Tabela 5, que mostra os passos executados após a escolha do procedimento a ser realizado pelo operador.

A função *iniciarProcedimento* dá início ao procedimento escolhido pelo operador para ser realizado, seguindo a inicialização do procedimento, o seguinte passo, a função *escolherAção*, que é usada no modo de treinamento manual e sequencial, é utilizada para determinar qual ação o operador deseja executar, verificando as ações possíveis que foram definidas para o procedimento, que depois é executada pela função *executarAção*. No modo automático, em vez de *escolherAção*, são usadas as funções *escolherAções* e *reverAções*, que são utilizadas, respectivamente, para executar a lista de ações determinadas para procedimento e para permitir repetir ações que usuário deseje rever.

Tabela 5 - Passos para a execução dos procedimentos.

Descrição	Função
iniciarProcedimento(..)	Iniciar procedimento escolhido pelo operador no cenário de treinamento.
escolherAção(..)	Modo de Avaliação: Identifica o objeto que o operador deseja manipular, e qual ação será executada no cenário em específico.
executarAções(..) reverAções(..)	Modo Automático: O operador visualiza as ações executadas automaticamente, com a possibilidade de revê-las quantas vezes forem necessárias para sua compreensão.
executarAção(..)	Executar a ação determinada pelo operador ou acontecerá à execução automática pela aplicação.
repetirOperação(..) reverAções(..)	Repetir o procedimento em específico para melhor absorção do conteúdo. Disponível para os dois modos de operação.
concluirOperação(..)	Concluir procedimentos, o operador terá a liberdade de encerrar o procedimento quando ele achar que finalizou a operação corrente.
cancelarAção(..) repetirOperação(..)	Cancelar operação. O operador terá a liberdade de cancelar/ refazer o procedimento em execução, quando ele achar que se precipitou em alguma ação e retornar ao menu de ações

Fonte: (SANTOS, 2013).

Após a execução de todos os passos o operador tem a opção de repetir o procedimento, executando novamente com *repetirOperação*, que funciona de modo semelhante à *reverAções*, mas sendo usada para os modos manuais e sequenciais. Em qualquer momento da operação o operador pode realizar o seu encerramento, que é realizado por *concluirOperação* e *cancelarOperação*, sendo que, com *cancelarOperação* o operador não é avaliado.

4.2. Cenários e Interação com o Usuário

O ATreVEE 3D tem como meios de entrada os dispositivos padrão, mouse e teclado. Portanto, todos os passos dos procedimentos e a interação com o sistema são executados com o uso de tais dispositivos. O simulador possui cinco diferentes cenas: três são executadas apenas no modo sequencial e duas que são executadas em todos os modos, mas só é realizada uma delas por treinamento, já que são as duas versões do mesmo procedimento (linha ativa e desativada) e são divididas em três etapas, uma para cada isolador.

Os três cenários exclusivos do modo sequencial são compostos pela chegada do operador à empresa, pela escolha de equipamentos e pela saída da empresa. A chegada e saída do operador são realizadas utilizando apenas comandos via teclado, com as teclas ‘W’, ‘A’, ‘S’ e ‘D’ ou setas direcionais para locomover a personagem pelo ambiente virtual, e com a tecla ‘E’ para abrir a porta da sala de equipamentos. A cena de escolha de equipamentos

possui a maior liberdade de interação, pois apresenta mais de um objeto interativo (equipamentos). Na sala de equipamentos é possível controlar a personagem com visão em primeira pessoa, mostrando apenas um braço, como pode ser visto na Figura 15, usando mouse e teclado como meios de entrada. Nesta cena, o objetivo do operador é a escolha e recolhimento dos equipamentos corretos para a operação de manutenção, a definição de com qual objeto interagir é realizada através da colisão com o modelo da mão da personagem e usando uma tecla do teclado para executar a ação de pegar.

Figura 15 - Cena de escolha de equipamentos no ATreVEE 3D.



Fonte: PrintScreen do uso do ATreVEE 3D.

As duas cenas restantes são as trocas de isoladores na linha ativa e linha desativada, sendo ambas divididas em três etapas. Cada etapa trata da troca de um dos três isoladores da cruzeta, mostrando os erros do usuário ao fim de cada etapa. Antes de iniciar o treinamento o usuário deve escolher o modo de treinamento e qual procedimento quer realizar, o na linha ativa ou na linha desativada, portanto, só uma das versões é realizada por seção de treinamento. Na cena da troca de isoladores no simulador a execução do procedimento é feita através da escolha da ação que se deseja executar via botões na interface gráfica, como podem ser vistos na Figura 16, e em seguida, a ação é mostrada por meio de animação. Os botões são representações dos objetos com os quais se deseja interagir, quando o objeto escolhido é feita a verificação de qual ação está disponível para ele, de acordo com que passos do procedimento já foram realizados.

Figura 16 - Escolha da ação com o cursor.



Fonte: *PrintScreen* do uso do ATreVEE 3D.

Capítulo

5

ATreVEE IN

O presente trabalho apresenta a integração de interação natural em simuladores de procedimento para o setor elétrico usando como caso de uso o ATreVEE 3D, desenvolvido por Santos (2013), e realização testes com voluntários do setor, das duas versões do simulador.

Com a adição da interação natural na nova versão do simulador, chamada ATreVEE IN, buscou-se um avanço na imersão dos simuladores, procurando formas intuitivas de interagir com o ambiente virtual, que se assemelhem às ações reais. Visando o auxílio ao treinamento de profissionais para que com uma melhor preparação, possam exercer suas atividades com maior segurança evitando a ocorrência de acidentes, diminuindo assim as fatalidades e os danos físicos e financeiros. Para isso, foi proposto utilizar um dispositivo que permita o uso de gestos como entrada, o controle Leap Motion. Com o uso de tal dispositivo visou-se um treinamento que se aproxime mais do real, procurando uma interação com o simulador que se assemelhe a execução real do procedimento.

5.1. Aplicação Piloto

Com a intenção de analisar a viabilidade da proposta foi desenvolvida uma aplicação piloto com o controle Leap Motion como meio de entrada, com a utilização de um cenário mais simples presente no simulador. Como mostrado na Figura 17, o cenário trabalhado é uma preparação para a atividade de manutenção, em que o operador deve escolher os equipamentos que serão necessários para a realização da manutenção que se encontram em uma sala dispostos em armários e bancadas, e o usuário deve “pegar” os itens corretos.

A aplicação piloto tem como meio de entrada o controle Leap Motion, por meio do qual é feita a interação com o cenário virtual utilizando os dados fornecidos do rastreamento das mãos. Como pode ser visto na Figura 17, o aplicativo possui dois modelos 3D que

representam as mãos rastreadas, com esses modelos é possível saber com que parte do cenário está sendo realizada a iteração.

Figura 17 - Aplicação piloto.

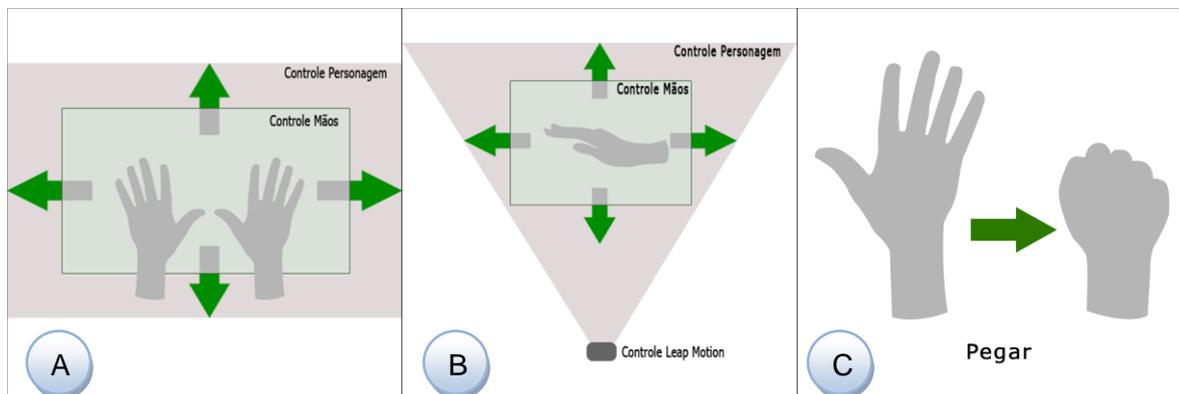


A atividade a ser realizada nesse cenário piloto, como foi dito anteriormente, é a escolha dos equipamentos que são necessários para a manutenção das linhas de transmissão, para pegá-los no cenário virtual deve-se tocá-los, usando o modelo como referência de onde se encontra a mão rastreada dentro do cenário virtual, e fechar a mão. Após pegar todos os itens da cena necessários a cena é finalizada.

A navegação pelo cenário é realizada totalmente a partir do rastreamento da mão, para locomover o “jogador”, e conseqüentemente as mãos e a câmera, para frente basta mover uma mão ou as duas em direção ao monitor, enquanto que para mover para trás deve-se mover a(s) mão(s) na direção contrária. Para evitar que todo movimento mova também a personagem e não só as mãos virtuais foram determinados limites que indicam em que área do campo de visão do dispositivo o rastreamento será interpretado como movimento da personagem, como pode ser visto na Figura 18a e na Figura 18b. Caso esses limites não sejam ultrapassados, o rastreamento das mãos movimentará apenas as mãos virtuais e os objetos com os quais elas estão interagindo. É possível também a rotação da personagem, tanto horizontalmente como verticalmente, sendo que na direção vertical o ângulo de rotação é limitado, para permitir apenas uma melhor visualização quando necessário. De modo semelhante ao movimento, a rotação é realizada com limites nas coordenadas X e Y do rastreamento, enquanto o movimento utiliza da coordenada Z. Nesse cenário piloto o gesto

utilizado é o de fechar a mão, que é mostrado na Figura 18c, que deve ser usado nas ferramentas expostas no ambiente, para pegá-las.

Figura 18 - Controles da aplicação piloto: (a) visão superior do controle Leap Motion; (b) visão lateral do uso do controle Leap Motion; (c) gesto utilizado para pegar os objetos na cena.



Com a construção do piloto, foram vistos os pontos em que se deve focar no modelo apresentado na Figura 14. No desenvolvimento da nova versão do simulador foi visto a necessidade de focar na construção de um módulo para fazer a interface com o dispositivo, usando os dados para interação com o ambiente virtual. No cenário usado para a aplicação piloto foi necessário também modificar os objetos para assim podê-los usar com a nova interface e possibilitando a alteração futura da interface utilizada com maior facilidade.

5.2. Funcionalidades

As etapas do treinamento foram divididas em três tipos de cena: Cena de Transição, Cena de Preparação e Cena de Manutenção. No ATreVEE IN as cenas de manutenção correspondem aos principais procedimentos: a troca de isoladores de poste trifásico na linha ativa e na desativada. O simulador apresenta também três outras cenas mais simples, duas delas cujo objetivo é apenas o deslocamento de um ponto a outro do ambiente virtual, essas são cenas de transição. A cena de escolha de equipamentos tem como funcionalidades a locomoção da personagem e a ação de pegar equipamentos.

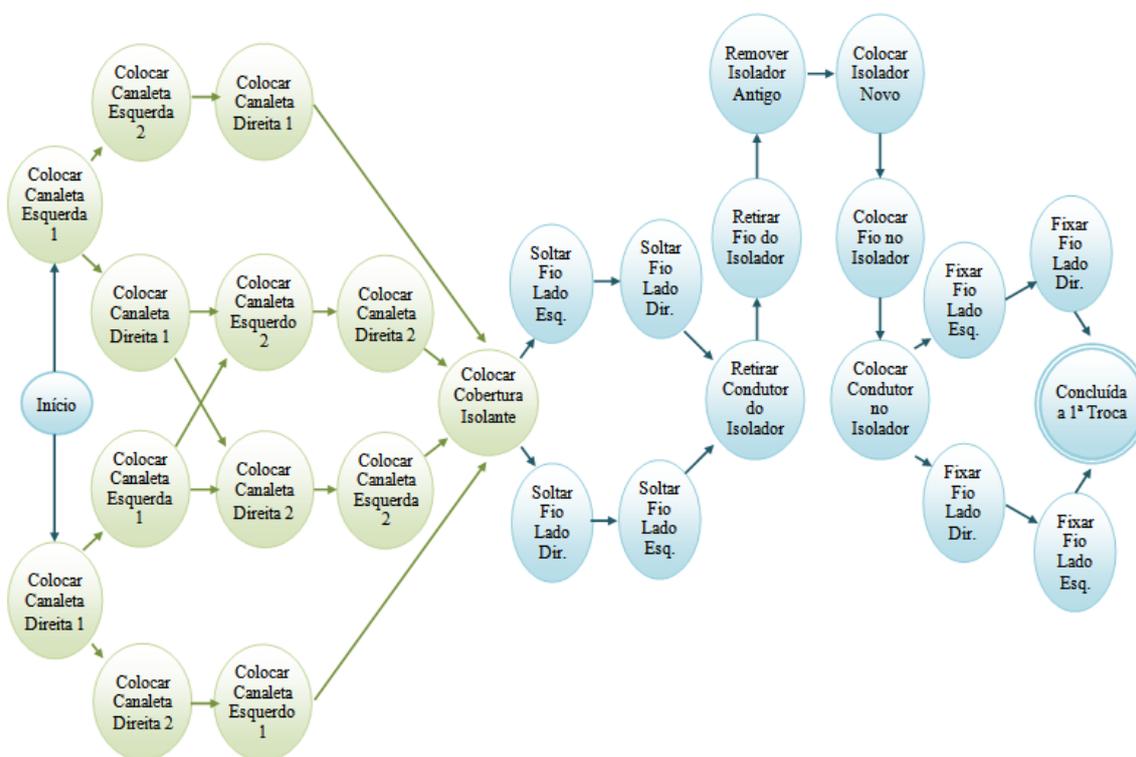
Na cena de manutenção, uma maior quantidade de funcionalidades deve estar presentes para a realização das atividades necessárias, para a troca do primeiro isolador, os seguintes passos devem ser realizados:

- Primeiramente o usuário deve colocar os canaletas protetoras nos primeiro e segundo condutores, colocando as dos fios mais próximos primeiro, seguido da cobertura isolante que é posta na cruzeta entre os dois isoladores;

- Depois deve ser liberado o condutor que é fixado com um fio de cobre que é preso no isolador, após soltar as duas pontas do fio do condutor, o condutor deve ser levantado (pegando pela canaleta) e posto em cima da cobertura na cruzeta;
- Então, se tira o fio de cobre do isolador e remove o isolador antigo da cruzeta, substituindo-o pelo novo;
- Com o novo isolador posto, deve ser colocado o fio de cobre no novo isolador e colocar o condutor em cima do isolador;
- Com o condutor em sua posição, deve-se fixá-lo prendendo-o com o fio de cobre.

Na Figura 19 é apresentada a máquina de estados que define a sequência correta de passos para a troca do primeiro isolador na linha ativa, destacando em verde os passos que devem ser ignorados para se ter os passos da troca na linha desativada.

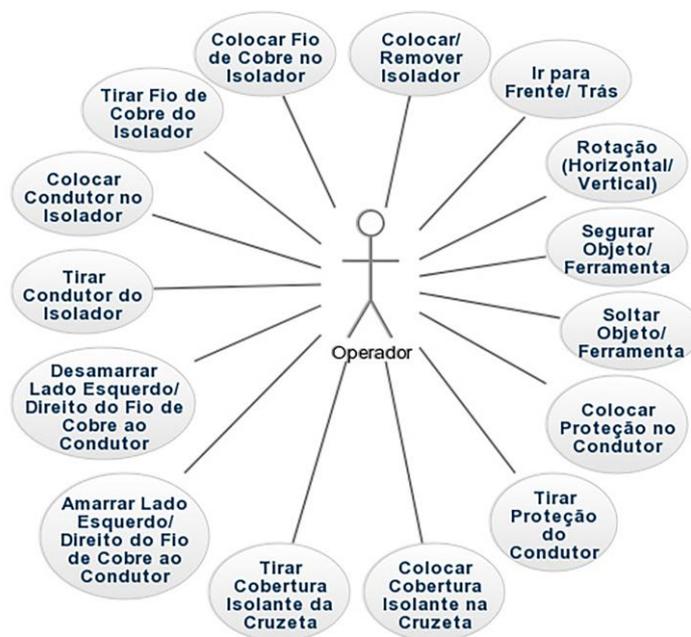
Figura 19 – Máquina de estados para a troca do 1º isolador em linha ativa, com os passos desnecessários para a linha desativada destacados em verde.



A partir do procedimento descrito acima, foi então projetado o diagrama de casos de uso apresentado na Figura 20, usado como guia para o desenvolvimento das funcionalidades necessárias. Como visto no diagrama as funcionalidades permitidas ao operador no cenário principal são: mover para trás e frente, rotacionar a câmera horizontal e verticalmente, segurar e soltar objeto ou ferramenta, colocar e tirar proteção do condutor e da cruzeta,

amarrar e desamarrar fio de cobre do condutor, colocar e tirar condutor do isolador, colocar e tirar fio de cobre do isolador e colocar e tirar isolador da cruzeta.

Figura 20 – Diagrama de casos de uso para a manutenção em linha ativa.



5.3. Desenvolvimento

Nesta seção serão detalhados alguns aspectos do desenvolvimento para melhor entender o funcionamento do sistema e as diferenças da implementação com IN. Durante o desenvolvimento desse trabalho, teve-se acesso ao projeto do ATreVEE 3D desenvolvido em (SANTOS, 2013) e foram utilizados na construção das cenas do ATreVEE IN os mesmos modelos tridimensionais que foram criados para a versão anterior.

5.3.1. Ferramentas Utilizadas

Nessa seção são mostradas as principais ferramentas utilizadas no desenvolvimento do ATreVEE IN, que foram elas o motor de jogos Unity 3D e o Leap Motion SDK.

5.3.1.1. Unity 3D

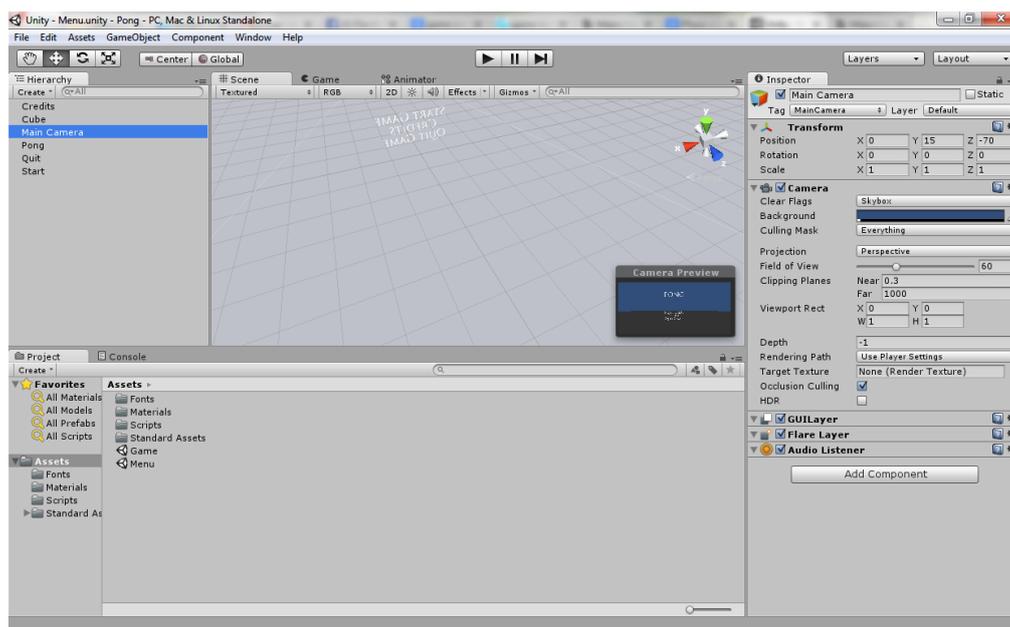
Com o grande aumento da complexidade dos jogos eletrônicos se tornou comercialmente inviável produzir todo o código de cada novo jogo. Em vez disso passou-se a utilizar motores de jogos, que Bianchini *et al.* (2006) define como “um conjunto de componentes de software integráveis e reutilizáveis, que fornecem serviços utilizados em um jogo eletrônico”. Esses componentes ajudam a construir os ambientes virtuais que compõem

os jogos, disponibilizando soluções já implementadas para renderização, inteligência artificial, entrada do usuário, colisão e simulação física, entre outras, que podem ser utilizadas para auxiliar a construção de qualquer ambiente virtual.

Para o trabalho desenvolvido foi escolhido o motor de jogos Unity 3D (UNITY, 2014) desenvolvido pela *Unity Technologies*, pois permite o desenvolvimento multi plataforma e multilinguagem, além de possuir uma versão livre da plataforma.

A Unity disponibiliza diversos *assets* na sua loja (pagos ou não) juntamente com *assets* disponibilizados por outros usuários para auxiliar no desenvolvimento, sendo *assets* todo o conteúdo que forma o jogo, como *scripts*, texturas e modelos. O motor Unity 3D ainda simplifica a criação de jogos através de uma ferramenta de edição, como mostrado na Figura 21, que permite adição de elementos ao ambiente de modo simples (arrastar e soltar), elementos esses que podem ser pré-fabricados como, chuva ou um terreno arborizado. Unity permite também a adição de propriedades físicas aos objetos por meio da interface do editor e a utilização de *scripts*, escritos em C#, JavaScript e Boo, para definir comportamentos para os objetos. Por exemplo, o que acontece com o objeto quando selecionado pelo jogador.

Figura 21 - Interface do editor do Unity 3D.



Fonte: *PrintScreen* da interface do Unity 3D.

5.3.1.2. Leap Motion SDK

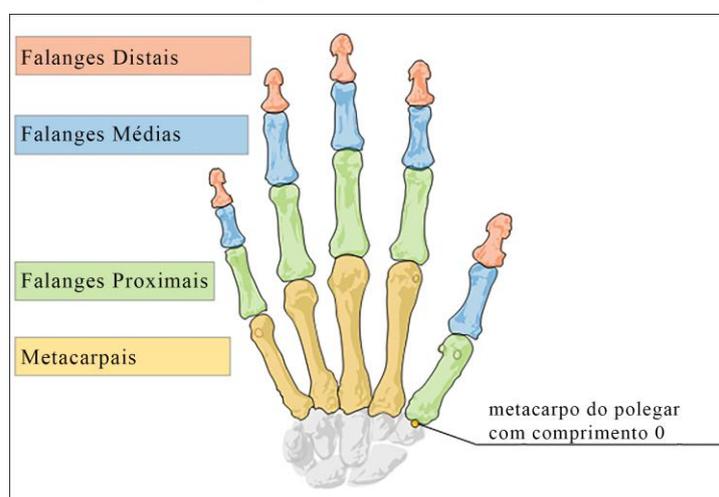
Para o desenvolvimento foi utilizado, juntamente com o motor Unity 3D, a versão 2.2.2 do Leap Motion SDK (Kit de Desenvolvimento de Software), que é fornecido pela Leap Motion para o desenvolvimento com o seu controle de movimento (LEAP MOTION, 2013).

Para melhor entender alguns aspectos do desenvolvimento mostrados nessa seção, é necessário fazer um rápido resumo de como são representadas as mãos e dedos a partir da documentação do Leap Motion SDK (LEAP MOTION, 2014).

A representação da mão no SDK disponibiliza informações sobre a identidade, posição entre outras características da mão detectada como, a lista de dedos associados com a mão. O controle fornece informações sobre cada dedo da mão, caso um dedo ou parte dele não está visível para o controle, as suas características são estimadas baseadas em observações recentes e em um modelo anatômico da mão humana mantido pelo SDK.

Um objeto do tipo *Finger* (dedo) fornece um objeto *Bone* (osso) para cada osso anatômico do dedo, sendo cada dedo composto por 4 ossos, como mostrado na Figura 22. Para o polegar é considerado um metacarpo de comprimento zero, já que ele possui apenas três ossos, dessa forma o polegar mantém a mesma estrutura que os demais dedos, facilitando a programação.

Figura 22 - Ossos da mão.



Fonte: Tradução de imagem obtida na documentação do Leap Motion SDK (LEAP MOTION, 2014).

Para auxiliar no desenvolvimento no Unity 3D com o Leap Motion, a empresa disponibiliza na *Asset Store* da Unity um conjunto de *Assets* (recursos) chamado *Leap Motion Core Assets* (Recursos Essenciais para o Leap Motion) (LEAP MOTION, 2015). Para esse trabalho foi utilizada a versão 1.0 desse conjunto de recursos.

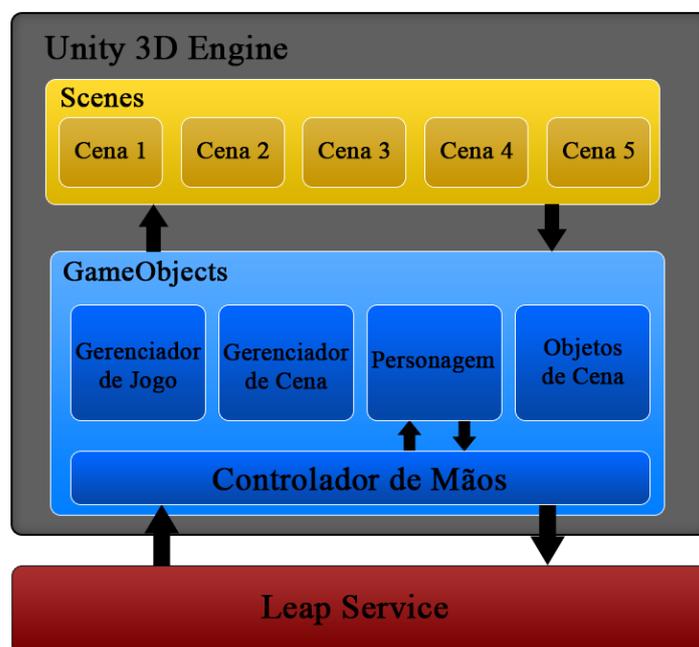
5.3.2. Arquitetura

A arquitetura do ATreVEE IN, apresentada na Figura 23, é dividida em duas camadas bases, *Leap Service* e *Unity 3D Engine*. A camada *Leap Service* é responsável pelo recebimento dos dados do dispositivo Leap Motion e a interpretação deles para o uso na

aplicação. Já a camada *Unity 3D Engine* engloba as soluções fornecidas pelo motor de jogos Unity 3D, todas as cenas construídas e também as operações e objetos desenvolvidos. Tudo o que o Unity 3D Engine engloba é agrupado em duas camadas distintas: a *Scenes*, que é composta das cenas que compõem o simulador, e a *GameObjects* que são os objetos usados para a composição das cenas, estes são divididos como cinco componentes.

O *Controlador de Mãos* é um componente da camada *GameObjects* responsável pelo recebimento dos dados interpretados na *Leap Service* e uso destes dados no ambiente virtual. Esse componente é encarregado também de representar e controlar as mãos no ambiente virtual, a partir dos dados recebidos. *Personagem* é a representação do operador no jogo, esse componente utiliza as informações colhidas pelo *Controlador de Mãos* para realizar a interface com a personagem via o rastreamento das mãos. Os *Gerenciador de Cena* e *Gerenciador de Jogo* são componentes responsáveis por controlar as atividades no simulador, o primeiro, é responsável por ações de controle na cena durante o procedimento, pela discriminação das etapas para a conclusão da cena e pelo monitoramento e avaliação das ações realizadas pelo usuário durante o treinamento, e o *Controlador de Jogo* é responsável pela iniciação da mudança de cena e pela escolha de cena e treinamento que se deseja realizar. Por último tem-se os *Objetos de Cena* que é o conjunto de objetos interativos que compõem a cena, com a definição geométrica e comportamental deles.

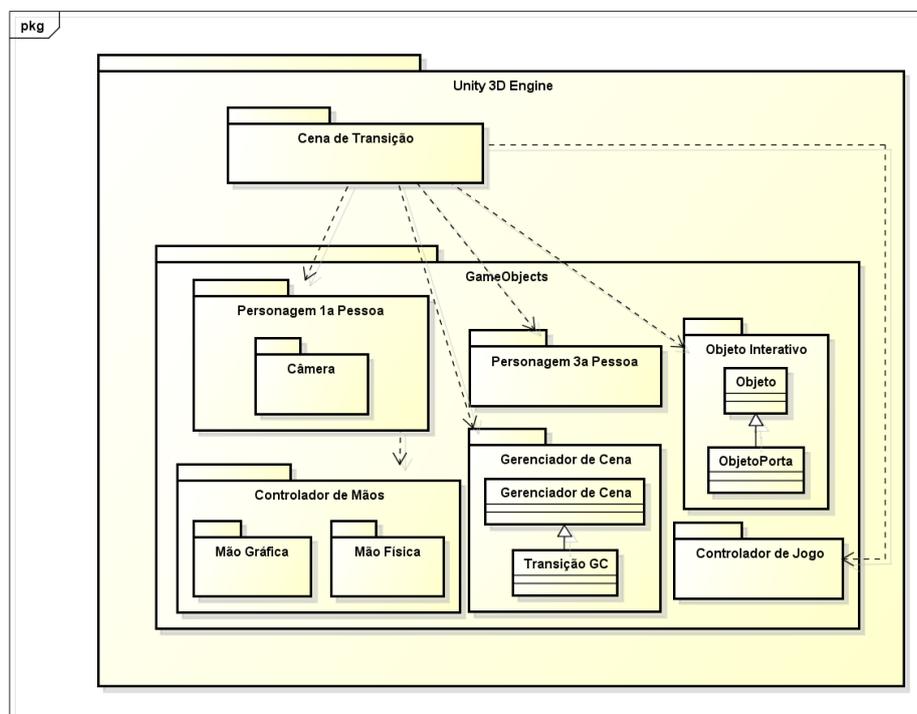
Figura 23 - Arquitetura do ATreVEE IN.



A seguir são apresentados diagramas de pacotes para os três tipos de cenas apresentados no simulador. Na Figura 24 é apresentado o diagrama para as cenas de

transição. Essa cena possui duas diferentes personagens que podem ser alternados durante a execução, uma em 1ª pessoa, usando o controle Leap Motion como meio de interação através do *Controlador de Mãos*, e outra em 3ª pessoa, com o teclado como meio de entrada. Como se pode perceber no diagrama, essa cena necessita apenas de um objeto interativo, o *ObjetoPorta*, esse objeto tem como função possibilitar a entrada da personagem na sala de equipamentos.

Figura 24 - Diagrama de pacotes para cena de transição.

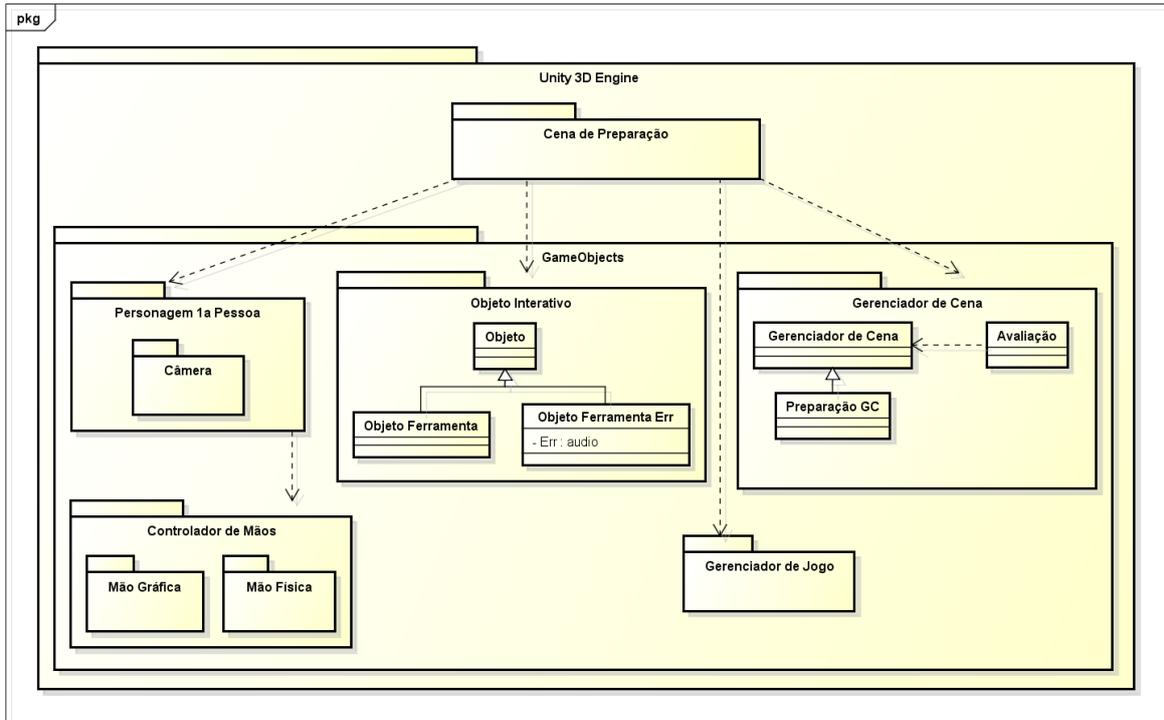


powered by Astah

No diagrama da cena de preparação, apresentado na Figura 25, pode-se ver que não se usa personagem em 3ª pessoa, é disponibilizada apenas a interação via controle Leap Motion e são usados dois tipos de objetos interativos: as ferramentas corretas e ferramentas erradas.

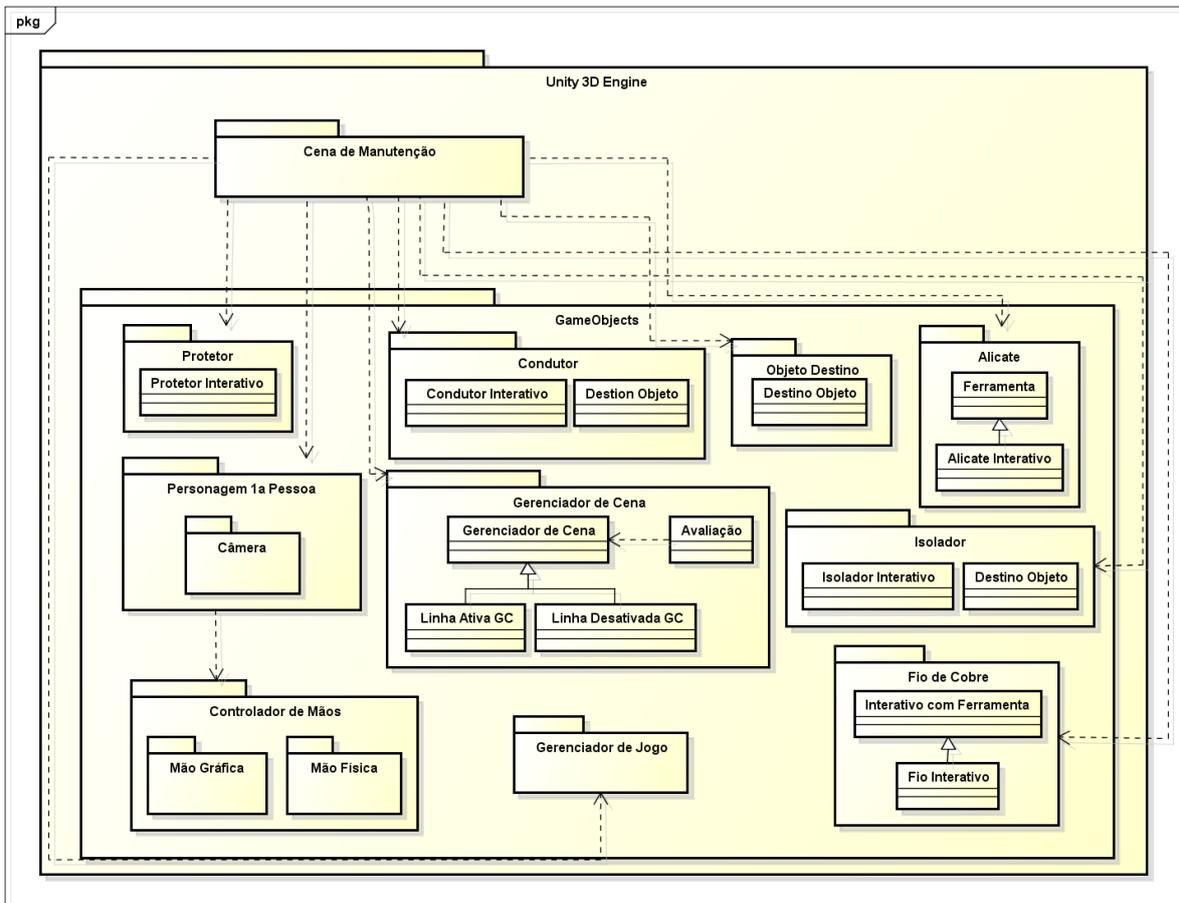
Nas duas cenas de manutenção, apresentadas no diagrama da Figura 26, é usado um maior número de objetos interativos, isso ocorre, pois elas representam um procedimento mais complexo do que as das demais cenas. Cada tipo de objeto desses possui uma definição diferente para representar comportamento e funcionalidade. Por exemplo, os objetos do tipo *Objeto Destino*, são objetos que tem como função o acoplamento de outros objetos em si, um exemplo desse tipo de objeto seria o parafuso, no qual é posto o isolador, e a cruzeta, na qual é colocada a cobertura isolante. Outros objetos que merecem destaque é a ferramenta *Alicate*, que é utilizada para controlar outros objetos, e o *Fio de Cobre*, com o qual só se pode interagir utilizando a ferramenta alicate.

Figura 25 - Diagrama de pacotes para cena de preparação.



powered by Astah

Figura 26 - Diagrama de pacotes para cena de manutenção.



powered by Astah

Como mostrado nas Figura 24, Figura 25 e Figura 26, todas as cenas apresentam os seguintes componentes: o *Gerenciador de Jogo*, que é o mesmo em todas as cenas e controla a alternância entre cenas e tipo de treinamento sendo realizado, o *Personagem em 1ª Pessoa*, o *Controlador das Mãos* (estes dois últimos são controlados através do controle Leap Motion, pelo usuário, para navegar pelo ambiente e interagir com os objetos), e o *Gerenciador de Cena*, que é definido para cada cena para cumprir necessidades diferentes, é neste componente que são definidos as regras da cena, e nas cenas de preparação e de manutenção é responsável também pela avaliação do desempenho do usuário.

5.3.3. Construção da Cena

O projeto no Unity 3D é composto por cenas (*scenes*), nas quais são montadas os ambientes virtuais, com suas ações, objetos e propriedades. Na Figura 27 e na Figura 28 são apresentadas as composições das cenas de troca do isolador no ATreVEE 3D e ATreVEE IN, respectivamente. A troca dos isoladores no ATreVEE 3D é feita em 3 cenas diferentes, sendo uma para cada isolador, e na Figura 27 pode-se ver a troca do primeiro deles na linha ativa. Como pode ser visto no esquema da cena do ATreVEE 3D, ela é composta pelos objetos de plano de fundo, modelos geométricos nos quais não se pode executar ações, e o objeto *Cena Manutenção 01*. Este objeto é composto pelos modelos geométricos do poste, das ferramentas e das mãos do operador, pelas animações dos passos do procedimento, e pelo *script* de controle do objeto. Neste *script* são criados botões na interface gráfica, para representar os objetos com os quais se quer interagir, e é iniciada a animação correspondente quando pressionado determinado botão.

Figura 27 - Cena de manutenção ATreVEE 3D.

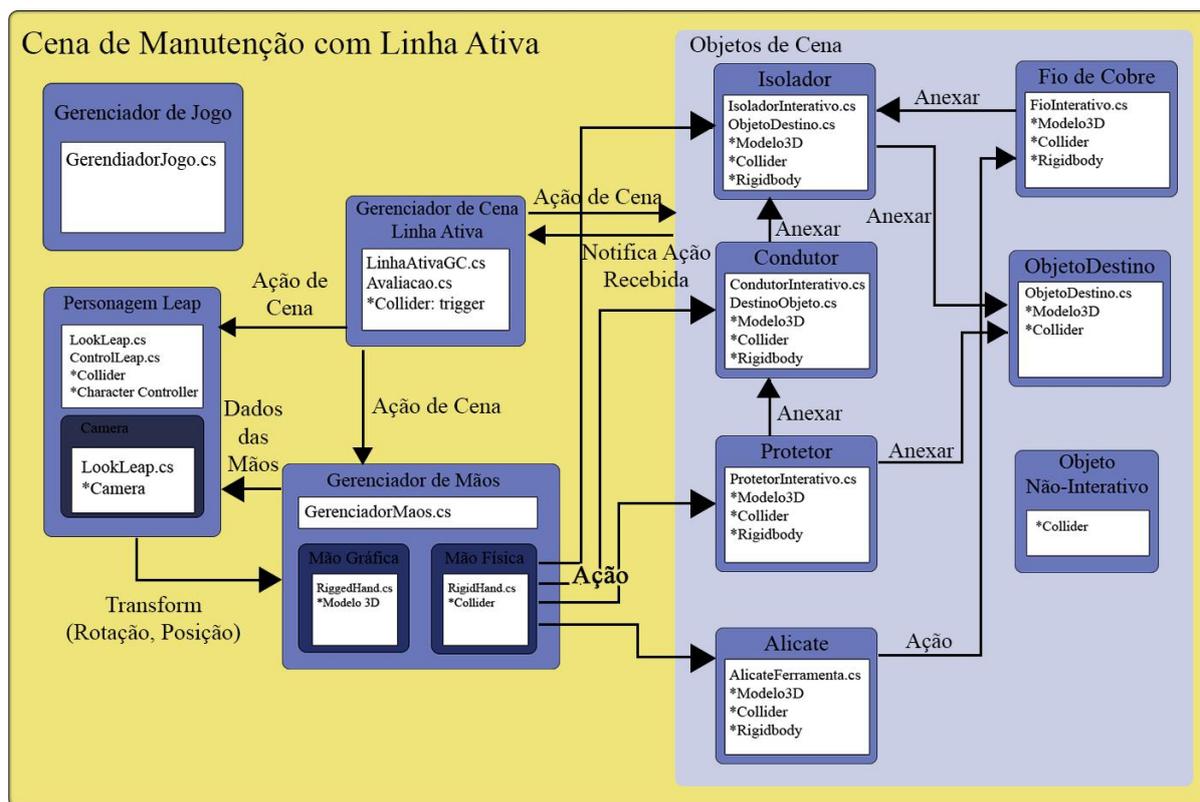


Para o novo simulador, buscando usufruir de toda a liberdade que o uso de interação natural fornece, a construção da cena foi feita de forma diferente: dividiu-se a cena em objetos independentes, para que se pudesse desenvolver comportamentos distintos de acordo com as necessidades exigidas pelas suas contrapartidas reais. A interação via o controle Leap

Motion é realizada através dos objetos *Gerenciador de Mãos* e *Personagem Leap*. O primeiro é responsável pelo recebimento dos dados do dispositivo, já interpretados pelo SDK, e pelo seu uso para representar as mãos na cena, permitindo o controle dos modelos das mãos e a interação com os objetos de cena. O *Personagem Leap* usa os dados recebidos pelo *Gerenciador de Mãos* para controlar a navegação da personagem, usando esses dados para a movimentação da personagem e para mudança de seu campo de visão através de rotação dela e da câmera.

Os ambientes virtuais das cenas de manutenção são compostos por um maior número de objetos estáticos e interativos. Os objetos estáticos são os que formam o ambiente, mas não há necessidade de interagir com eles, um exemplo seria a base do poste e os modelos de casas que se encontram no solo do ambiente. Os objetos interativos são aqueles que possuem comportamentos definidos para reagir à interação da personagem. As cenas de manutenção, como mencionado na seção anterior, possuem dois gerenciadores: o de jogo, que é o mesmo para todas as cenas do simulador, e o de cena, que define as regras da cena e analisa como elas foram cumpridas, como por exemplo, metas a serem realizadas. A máquina de estados da Figura 19 foi utilizada como base para o desenvolvimento do gerenciador, para definir as regras e analisar o desempenho do operador.

Figura 28 - Cena de manutenção ATreVEE IN.

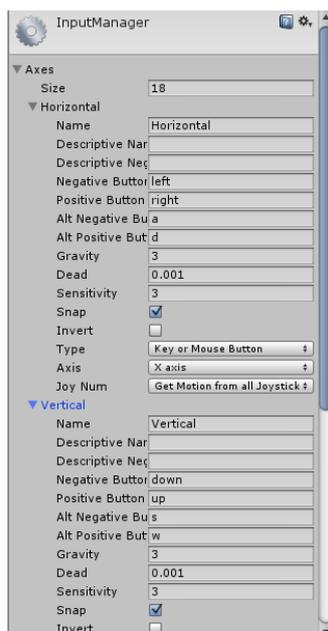


5.3.4. Navegação

A navegação da personagem é realizada de forma semelhante à versão com mouse e teclado, ambas as soluções usam o *CharacterController*, um *asset* da Unity. O *CharacterController* auxilia o uso de uma personagem no jogo, disponibilizando a sua movimentação já restringida por colisões, utilizando uma cápsula, para representar o seu corpo para as colisões.

Juntamente com o *CharacterController* há uma solução para a movimentação de uma personagem em primeira pessoa (ponto de vista da personagem) que utiliza uma cápsula como representação gráfica, como pode ser visto na Figura 62, para auxiliar o desenvolvedor no editor de cena do Unity 3D. Nessa solução são utilizados os eixos de entrada que podem ser configurados na interface gráfica do gerenciador de entrada (*input manager*) do Unity 3D, para obter a direção de movimento. Portanto, são utilizados os eixos “Horizontal” e “Vertical”, mostrados na Figura 29, o primeiro usa ‘A’ e a seta para a esquerda como botões de entrada de valor negativo e ‘D’ e a seta para a direita como botões positivos, e o segundo tem como entrada negativa as teclas ‘S’ e seta para cima, e como entrada positiva ‘W’ e seta para baixo.

Figura 29 - Gerenciador de entrada e eixos horizontal e vertical.



Na nova versão do simulador foi alterado o tipo de entrada utilizada, pegando a posição das mãos do usuário que está sendo rastreada pelo controle Leap Motion, portanto foi necessário modificar o controle da personagem para aceitar tais dados de entrada. Como foi mostrado na Figura 18, a direção do movimento é identificada a partir da posição das mãos

em relação a uma caixa delimitadora na área visível do dispositivo. Quando uma das mãos passa a delimitação frontal ou traseira, a personagem se move para frente ou para trás, respectivamente. Na Figura 30 vê-se o código fonte da função *Update* (que é executada em todo quadro – *frame*) da solução para o uso do teclado, em que é obtido o valor da entrada em cada eixo através dos comandos *Input.GetAxis("NomeDoEixo")*, esse valor é então usado para a construção de um vetor de direção, com o valor do eixo Horizontal como o valor X e o do eixo Vertical como o valor Z, como feito na linha destacada da Figura 30. Esse vetor de direção após ser normalizado e tratado para melhor controle é então enviado para o script responsável pela movimentação da personagem.

Figura 30 - Código fonte da definição da direção de movimento com o teclado.

```
// Atualização chamada uma vez por quadro (frame)
function Update () {
    // Recebe vetor de entrada do teclado
    var vetorDirecao = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));

    if (vetorDirecao != Vector3.zero) {
        // Pega o comprimento do vetor de direção e o normaliza
        var comprimentoDirecao = vetorDirecao.magnitude;
        vetorDirecao = vetorDirecao / comprimentoDirecao;

        // Ter certeza que a direção não é maior que 1
        comprimentoDirecao = Mathf.Min(1, comprimentoDirecao);

        // Faz o vetor de entrada mais sensível aos extremos e menos sensível aos valores intermediários
        comprimentoDirecao = comprimentoDirecao * comprimentoDirecao;

        // Multiplica a direção normalizada pelo comprimento modificado
        vetorDirecao = vetorDirecao * comprimentoDirecao;
    }

    // Aplica a direção calculada para o motor da personagem
    motor.inputMoveDirection = transform.rotation * vetorDirecao;
}
```

Fonte: *Script FPSInputController*, um *asset* disponibilizado no Unity 3D no conjunto de *assets CharacterController*.

Na Figura 31 é apresentado o código fonte da função *Update* da solução para o controle Leap Motion, no qual primeiramente é escolhida a mão que será responsável pelo movimento, sendo essa, a mão que está mais distante do centro da área de rastreamento. Com a mão escolhida, é gerado o vetor de direção através da função *movimentoEmZ*, apresentada na Figura 32.

Figura 31 - Código fonte da definição da direção de movimento com o controle Leap Motion.

```
// Atualização chamada uma vez por quadro (frame)
void Update () {
    Vector3 vetorDirecao = new Vector3();
    Mao maoEscolhida = null;
    maos = controladorMaos.getMaos ();

    foreach(ModeloMao h in maos.Values){
        //Escolha da mão dominante: mão mais distante do centro
    }

    if(maoEscolhida != null)
        vetorDirecao = movimentoEmZ(maoEscolhida);

    if (vetorDirecao != Vector3.zero) {
        // Pega o comprimento do vetor de direção e o normaliza
        float comprimentoDirecao = vetorDirecao.magnitude;
        vetorDirecao = vetorDirecao / comprimentoDirecao;

        // Ter certeza que a direção não é maior que 1
        comprimentoDirecao = Mathf.Min(1, comprimentoDirecao);

        // Faz o vetor de entrada mais sensível aos extremos e menos sensível aos valores intermediários
        comprimentoDirecao = comprimentoDirecao * comprimentoDirecao;

        // Multiplica a direção normalizada pelo comprimento modificado
        vetorDirecao = vetorDirecao * comprimentoDirecao;
    }

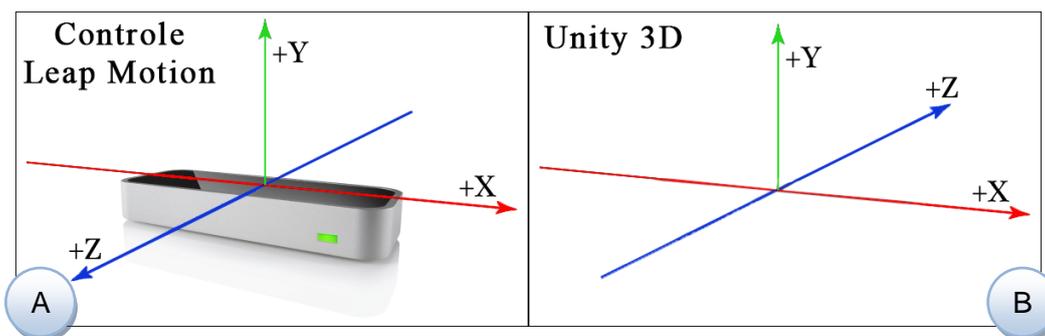
    // Aplica a direção calculada para o motor da personagem
    motor.inputMoveDirection = transform.rotation * vetorDirecao;
}
}
```

Figura 32 - Código fonte da definição da direção do vetor de direção com o controle Leap Motion.

```
public Vector3 movimentoEmZ (Mao m){
    Vector3 vetorDirecao = new Vector3();
    //Se estiver além da delimitação traseira, movimento no sentido positivo do eixo Z (trás)
    if(m.PosicaoMao.z - centroZ < - limiarZ)
        vetorDirecao = new Vector3 (0, 0, -(m.PosicaoMao.z-centroZ+limiarZ));
    //Se estiver além da delimitação frontal, movimento no sentido negativo do eixo Z (frente)
    else if(m.PosicaoMao.z - centroZ > limiarZ)
        vetorDirecao = new Vector3 (0, 0, -(m.PosicaoMao.z-centroZ-limiarZ));
    return vetorDirecao;
}
}
```

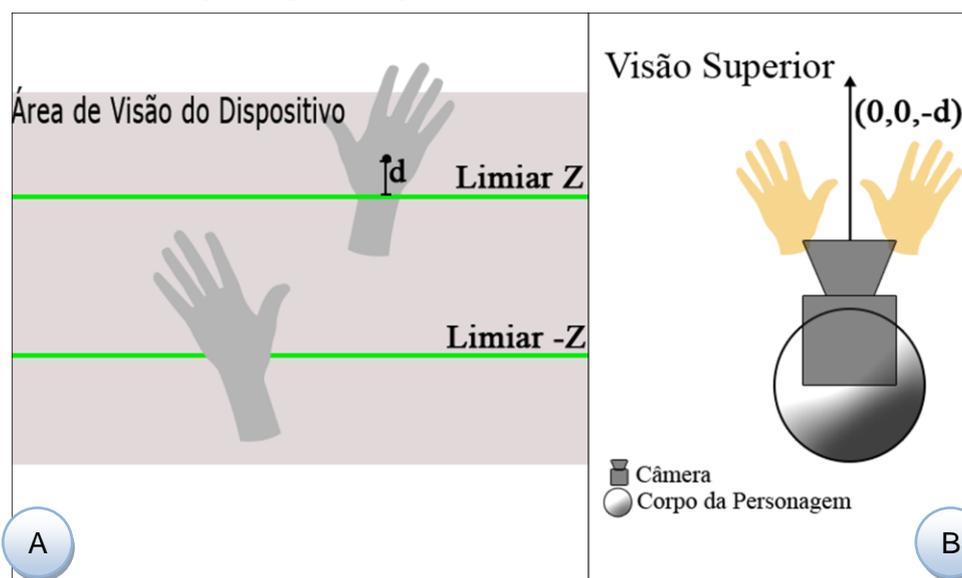
A função *movimentoEmZ*, apresentada na Figura 34, é responsável pela verificação da posição da mão, analisando se ela passou da delimitação frontal ou traseira, como disposto na Figura 34a. Caso isso tenha ocorrido, é usada a distância da mão em relação ao limiar, representada por 'd' na figura, para gerar o vetor de direção que será retornado. A distância é calculada usando somente os valores do eixo Z e seu valor, como mostrado na Figura 34b, é usado como o valor Z do vetor de direção criado. Já que os planos cartesianos do dispositivo e do ambiente possuem o eixo Z em sentidos contrários, como mostrado na Figura 33, o valor deve ser multiplicado pelo valor -1.

Figura 33 - Sistema de coordenadas do dispositivo Leap Motion (a) e do ambiente virtual (b).



Fonte: (a) Sistema de coordenadas do controle na documentação oficial da Leap Motion⁸.

Figura 34 - Movimento da personagem: (a) representação do movimento das mãos; (b) movimento resultante.



Além da movimentação da personagem a navegação é composta da rotação dela e da câmera. Essa rotação é realizada via o *script LookMouse* e *LookLeap*, para o uso do mouse e do controle Leap Motion, respectivamente. Ambos os *scripts* utilizam os dispositivos de entrada para rotacionar o objeto ao qual estão anexados, na construção das cenas esses *scripts* são utilizados para rotacionar toda a personagem em torno do eixo Y a partir do movimento no eixo X com os dispositivos e para rotacionar a câmera, juntamente com as suas mãos, em torno do eixo X a partir de movimentos no eixo Y.

Na Figura 35 vê-se a execução da rotação, via mouse, do objeto ao qual o *script* foi anexado, nela é possível ver que há três modos de rotação definidos nos blocos *if-else*. O primeiro bloco executa a rotação quando é desejada a realização nos dois eixos, o segundo faz a rotação apenas com o valor do mouse no eixo X e o terceiro bloco realiza a rotação com

⁸ Disponível em:

<https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Overview.html?proglang=current>.
Acesso em: Junho de 2015.

o valor do eixo Y. No controle da personagem foi usado o *script* para rotacionar com os valores do eixo X do mouse e no controle da câmera foi usada a rotação com os valores de do eixo Y. Essa escolha é feita após anexar os *scripts* aos objetos de cena na interface gráfica do Unity 3D, quando é feita a escolha em que eixo do mouse quer ser utilizado, sendo a câmera afetada pela rotação da personagem por ser um objeto filho do objeto personagem.

Figura 35 - Código fonte da rotação da personagem com o mouse.

```
// Atualização chamada uma vez por quadro (frame)
void Update ()
{
    //Se forem escolhidos ambos os eixos para rotacionar
    if (eixos == RotationAxes.MouseXAndY)
    {
        float rotacaoX = transform.localEulerAngles.y + Input.GetAxis("Mouse X") * sensibilidadeX;

        rotacaoY += Input.GetAxis("Mouse Y") * sensibilidadeY;
        rotacaoY = Mathf.Clamp (rotacaoY, minimoY, maximoY);

        transform.localEulerAngles = new Vector3(-rotacaoY, rotacaoX, 0);
    }
    //Rotação somente com o eixo X do mouse
    else if (eixos == RotationAxes.MouseX)
    {
        transform.Rotate(0, Input.GetAxis("Mouse X") * sensibilidadeX, 0);
    }
    //Rotação somente com o eixo Y do mouse
    else
    {
        rotacaoY += Input.GetAxis("Mouse Y") * sensibilidadeY;
        rotacaoY = Mathf.Clamp (rotacaoY, minimoY, maximoY);

        transform.localEulerAngles = new Vector3(-rotacaoY, transform.localEulerAngles.y, 0);
    }
}
}
```

Fonte: *Script LookMouse*, um *asset* disponibilizado no Unity 3D no conjunto de *assets CharacterController*.

Figura 36 - Código fonte da rotação da personagem com o controle Leap Motion.

```
// Atualização chamada uma vez por quadro (frame)
void Update () {
    if (gameObject.CompareTag ("Jogador") && !initiated) {
        transform.Rotate (rotInicialJogador);
        iniciado = true;
    }
    Mao maoEscolhidaY = null;
    Mao maoEscolhidaX = null;

    if (controladorMao != null) {
        maos = controladorMao.getMaos();
        if(maos == null || maos.Count < 1)
            return;

        foreach(ModeloMao m in maos.Values){
            if(h != null && h.GetLeapHand() != null){
                //Verifica qual mão possui o valor de X mais distante do centro
                //e qual possui o valor de Y mais distante do centro para definir
                //qual mão vai definir a rotação em cada eixo
            }

            //Se for para rotacionar com o valor de X executa a função rotacionaComX
            if(X)
                rotacionaComX(maoEscolhidaX);
            //Se for para rotacionar com o valor de Y executa a função rotacionaComY
            if(Y)
                rotacionaComY(maoEscolhidaY);
        }
    }
}
}
```

A função *rotacionaComX* apresentada na Figura 37, define o ângulo de rotação em torno do eixo Y utilizando os valores obtidos da movimentação da mão no eixo X e executa a rotação, quando não ultrapassar limites de ângulos definidos na construção da cena. Na Figura 37 é apresentada apenas a solução para os valores do eixo X, quando é ultrapassada a delimitação direita, o lado esquerdo e a solução para os valores do eixo Y foram realizados de forma semelhante ao apresentado. Na *rotacionaComX* é utilizada a distância da posição da mão no eixo X com o centro do eixo como ângulo de rotação, esse valor é suavizado por um valor definido na definição da cena para controlar a velocidade da rotação realizada. Após a definição do ângulo, é então verificado se foi definido valores limitantes para o ângulo, caso sim, a rotação só é executada se o ângulo estiver entre os dois valores fornecidos.

Figura 37 - Código fonte da função que define o ângulo de rotação a partir do controle Leap Motion.

```
//Rotaciona em torno do eixo Y com os valores do eixo X do dispositivo
private void rotacionarComX(Mao m){
    //Verifica se ultrapassou o delimitador direito
    if (m.PosicaoMao.x > limiarX) {

        //Distância entre valores X da delimitação ultrapassada e da posição da mão
        //utilizando um valor para suavizar a rotação
        float mudanca = ((m.PosicaoMao.x - limiarX) * suavizar);

        //Vetor com o valor do ângulo de rotação
        Vector3 angulo = new Vector3 ();
        angulo.y = mudanca;

        //Se houver restrição máxima e mínima dos ângulos de rotação
        if(limiteAnguloX){
            if(alemCircX){
                if(transform.rotation.eulerAngles.y >= minX-10 || transform.rotation.eulerAngles.y < maxX){
                    transform.Rotate(angulo);
                }
            }
            else {
                if(transform.rotation.eulerAngles.y >= (minX)-10 && transform.rotation.eulerAngles.y < (maxX))
                    transform.Rotate(angulo);
            }
        }
        else
            transform.Rotate (angulo);
    }
    //Verifica se ultrapassou o delimitador esquerdo
    else if (m.PosicaoMao.x < -limiarX) {
        float mudanca = ((m.PosicaoMao.x + limiarX) * suavizar);

        //Restante semelhante ao lado direito
    }
}
```

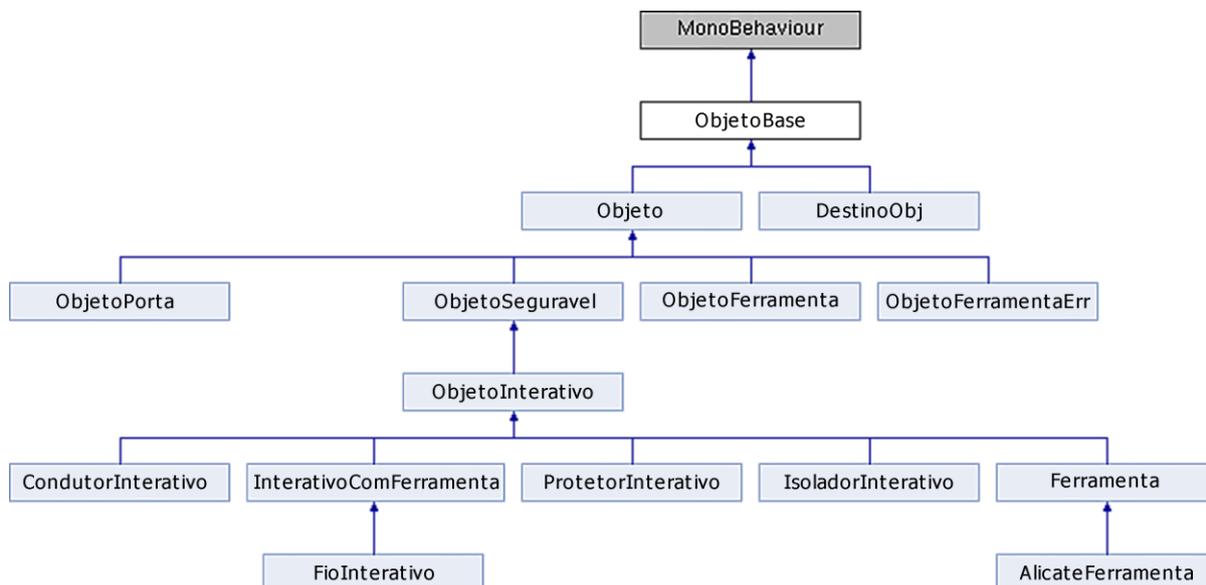
5.3.5. Objetos Interativos

No ATreVEE 3D, a cena de manutenção é composta por vídeos de animação, em que, com botões na interface gráfica, o usuário escolhe qual ação vai ser exibida no momento. Diferentemente, a cena de preparação proporciona um ambiente interativo que possui dois tipos de objetos interativos: ferramenta certa e ferramenta errada. Então para o simulador ATreVEE IN, que procura gerar ambientes interativos para todas as cenas disponibilizadas,

foi necessário o desenvolvimento de comportamentos para os objetos interativos de todas as cenas, usando o diagrama de casos de uso apresentado na Figura 20, como guia de quais funcionalidades os objetos deveriam possuir.

No ATreVEE IN foram desenvolvidos diferentes tipos de objetos, como apresentado na Figura 38, as subseções seguintes irão descrever brevemente cada tipo de objeto presente nas cenas.

Figura 38 - Classes definidas para os objetos interativos e suas relações.



5.3.5.1. Objeto Base

ObjetoBase é a classe que define algumas funções que são utilizadas por todos os objetos do simulador, como funções de selecionar e cancelar seleção e outras funções mais simples de requisição de informação armazenadas em variáveis privadas, como se o objeto pode ser selecionado.

Ao realizar a seleção, o objeto é realçado com um material que é disponibilizado pelo desenvolvedor na montagem da cena, adicionando o material de realce à lista de materiais do objeto em questão e removendo-o quando a seleção é cancelada. Essa seleção é usada para indicar ao usuário com qual objeto ele irá interagir ao realizar uma ação, sendo selecionado o objeto mais próximo da mão (estando ele perto o suficiente para interagir).

5.3.5.2. Objeto

A classe *Objeto* define o básico para os objetos interativos da cena, definindo as funções que são chamadas quando pega-se um objeto ou o solta. Essas funções aqui definidas são versões mais simples que são sobrescritas nas classes derivadas para melhor cumprir os

casos específicos de cada tipo de objeto. Na Figura 39 são mostradas as funções executadas ao pegar um objeto, vê-se que são versões bem simples dessas ações, em que é atualizado o estado do objeto para indicar que está sendo segurado, é adicionada uma referência da mão que o está segurando ao objeto e uma referência ao objeto na mão que o está segurando, e por último, é cancelada a seleção do objeto.

Figura 39 - Código fonte das funções *AoPegar* e *PegarObjeto* da classe *Objeto*.

```
public virtual void AoPegar (ModeloMao m)
{
    if (!dono) {
        PegarObjeto (m) ;
    }
}
public virtual void PegarObjeto (ModeloMao m) {
    estaSegurando = true;
    dono = m;
    m.setObjetoAtivo (this) ;
    DeSelect (h) ;
}
```

5.3.5.3. Destino de Objeto

São destinos de objeto, aqueles que permitem que outros objetos sejam anexados a eles, como o isolador, que permite serem anexados o condutor e o fio de cobre a ele. A *DestinoObj* também deriva da classe *ObjetoBase*, portanto também é selecionável quando se pode interagir com ele, a diferença nas duas seleções é feita com o uso de materiais diferentes, utilizando verde para o destino e amarelo para os demais, e na decisão de quando devem ser selecionados, que será visto na seção 5.3.5.8.

Cada objeto destino possui diversas listas: uma lista de objetos que estão anexados a ele, uma lista com os tipos de objetos que podem ser anexados a ele e uma lista com identificadores atribuídos aos itens, quando anexados. Essas listas fazem o controle de quando um objeto que está sendo segurado pode ser anexado através das funções *AcharPosiçãoLivre* e *PodeAnexar*, a primeira procura na lista de objetos anexados soma posição livre para o tipo de objeto que se quer anexar, e a segunda verifica se o destino aceita o tipo de objeto que se deseja anexar e se já não há outro objeto anexado.

Quando se deseja anexar um objeto a um destino é chamada a função *Anexar*, apresentada na Figura 40, em que é inicialmente feita uma busca de uma posição válida e livre na lista de objetos anexados, caso seja encontrada é obtido o identificador dessa posição. Com esse identificador é feita a anexação do objeto, primeiramente marcando a posição

como ocupada e depois enviando ao objeto anexado informações, como a rotação e posição que ele utilizará enquanto anexado, o identificador do objeto e que o objeto foi anexado.

Para desanexar o objeto é feita a busca do item que se deseja remover na lista de anexos, e se achado é realizado o processo inverso, limpando as referências que possam existir no destino e no objeto anexado e marcado a posição como livre e o objeto como não anexado.

Figura 40 - Código Fonte da função *Anexar* da classe *DestinoObj*.

```
public void Anexar(ObjetoInterativo obj){
    int objId = AcharPosicaoLivre(obj);

    if(objId > -1){
        //A posição do objId é marcada como ocupada
        anexado[objId] = true;

        //Referência ao objeto anexado
        objetosAnexados[objId] = obj;
        objetosAnexados[objId].estaAnexado = true;
        //Transform (posição, rotação, etc.) do objeto ao ser anexado
        objetosAnexados[objId].transformAnexado = transformsAnexados[objId];
        //Definição do id do objeto anexado
        objetosAnexados[objId].id = id[objId];
        //Id do objeto anexado foi definido
        objetosAnexados[objId].idDefinido = true;
        DeSelect (obj.dono);
        //Verifica se o destino já está todo preenchido, se sim, desabilita a seleção do destino
        AtualizarPodeRealcar();
    }
}
```

Na Tabela 6 são listados os destinos usados no desenvolvimento e os objetos aceitados por eles.

Tabela 6 - Relações entre destinos e objetos.

Destino	Objetos
Condutor	Protetor (Canaleta Protetora)
Isolador	Condutor e Fio de Cobre
Pino na Cruzeta	Isolador
Cruzeta (entre isoladores)	Protetor (Cobertura Isolante)

5.3.5.4. Objetos da Cena de Preparação

Para a cena de preparação, como feito no ATreVEE 3D, foram desenvolvidos dois tipos de objetos: *ObjetoFerramenta* e *ObjetoFerramentaErr*, para as ferramentas corretas e para as erradas, respectivamente. No ATreVEE 3D, esses objetos foram definidos como é apresentado nas Figura 41a e Figura 41b, a primeira mostra o tratamento da colisão com uma ferramenta correta e a segunda, com uma ferramenta errada. A interação com os dois tipos de

objetos é realizada quando o modelo da mão colide com um deles, ao ser realizada a colisão, quando o objeto é uma ferramenta correta é verificada se a tecla ‘E’ foi pressionada (no trecho: *if(acaoMaoDireita.nessE==1)*), se sim é adicionado um ponto ao gerenciador da cena (*proximaFase3*) e depois o objeto é destruído para removê-lo da cena. Já quando o objeto com o qual houve a colisão é uma ferramenta errada, é apenas executado um áudio para indicar o erro, como mostrado na Figura 41b.

Figura 41 - Parte do código fonte: (a) da ferramenta correta no ATreVEE 3D; (b) da ferramenta errada no ATreVEE 3D.

```

function OnCollisionEnter(collision: Collision )
{
    if(acaoMaoDireita.nessE==1)
    if( collision.gameObject.tag == "mao")
    {
        teclaE = 1;
        proximaFase3.ponto = proximaFase3.ponto+1;
        Destroy(gameObject);
    }
}

function OnCollisionEnter(collision: Collision )
{
    if( collision.gameObject.tag == "mao")
    {
        if(!audio.isPlaying){
            audio.clip = sErro;
            audio.Play();
            teclaErro=1;
            cond = 2;
        }
    }
    else{
        cond =1;
    }
}

```

A B

Fonte: Código fonte do projeto do ATreVEE 3D.

Utilizando a estrutura definida nas classes base, o desenvolvimento desses objetos no ATreVEE IN foi realizado como apresentado nas Figura 42a e Figura 42b. A primeira mostra a solução para a ferramenta correta, onde é reescrita a função *AoPegar* para chamar a função *Ação* (no código, chamada *Acao*), que por sua vez passa o comando de pegar cena para o gerenciador de cena onde será contabilizada a pontuação e removido o objeto da cena após ser pego. A sequência de passos da escolha do objeto é a seguinte: ao pegar o objeto que está selecionado (que contém uma referência na variável *aoAlcance* da mão) é chamada a função *AoPegar* que remove a referência da mão ao objeto selecionado, remove a seleção e depois chama a função *Acao* que executa a ação que se deseja ao pegar o objeto, no caso, informar ao gerenciador de cena que a ferramenta foi pego.

Para a ferramenta errada a *AoPegar* somente chama a função *Acao* já que não há a necessidade de remover a seleção e a referência, pois o objeto não será removido da cena e continuará ao alcance da mão. Então, a função chamada executa o áudio de erro e informa ao gerenciador de cena o erro cometido.

Figura 42 - Parte do código fonte: (a) da ferramenta correta no ATreVEE IN; (b) da ferramenta errada no ATreVEE IN.

```

public override void AoPegar(ModeloMao m){
    if (!dono)
    {
        m.aoAlcance = null;
        DeSelect(m);
        Acao();
    }
}

public override void Acao(){
    gerenciador_cena.PegaFerramenta(this);
}
}

```

A

```

public override void AoPegar(ModeloMao m){
    if (!dono)
    {
        Acao();
    }
}

public override void Acao(){
    if(!audio.isPlaying){
        audio.clip = somErro;
        audio.Play();
        gerenciador_cena.erroFerramenta(this);
    }
}
}

```

B

5.3.5.5. Objeto Segurável e Objeto Interativo

Os objetos seguráveis são aqueles que o usuário pode segurar, ao pegá-lo o modelo da mão é modificado por um segurando o objeto e o objeto tem seus valores de posição atualizados e sua renderização desativada, tornando o novo modelo da mão uma representação também do objeto. É nessa classe que é definida a versão da função *Acao* utilizada pela maioria dos objetos derivados, onde a função executa a alteração do modelo da mão pelo modelo composto e realiza a ocultação do objeto.

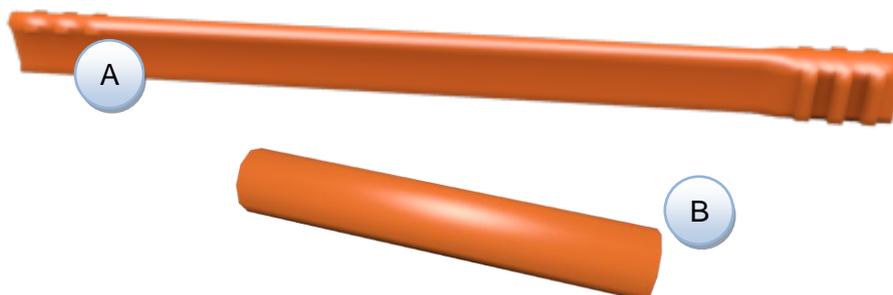
Os objetos do tipo interativo adicionam a anexação a outros objetos (destinos), essa anexação se dá da seguinte forma, quando o objeto está sendo segurado e um destino está selecionado, ao soltar o objeto este se anexa ao destino, redefinindo o modelo da mão para o padrão, chamando a função *Anexar* do destino e atualizando a posição e rotação para os valores que ele deve utilizar ao estar anexado, que são armazenados pelo objeto destino. A desanexação ocorre ao pegar um objeto que está anexado, realizando os procedimentos normais de pegar um objeto adicionado da chamada da função *Desanexar* do destino. Em ambas as ações, é informado ao gerenciador de cena qual ação foi realizada (anexar ou desanexar) para que o gerenciador possa monitorar e atualizar estados na cena.

5.3.5.5.1. Protetor

Objeto utilizado tanto nos condutores como na cruzeta, representando dois tipos de protetores diferentes, como apresentados na Figura 43, mostrando na Figura 43a o modelo da canaleta protetora utilizada no condutor, para permitir manuseá-lo com a linha ativa, e na Figura 43b o modelo da cobertura isolante, protetor que é colocado na cruzeta entre os isoladores, para descansar o condutor enquanto é feita a troca do isolador. A diferenciação

dos dois tipos é feita via etiquetas (*tags*) que são utilizadas na verificação de se o destino aceita o objeto segurado.

Figura 43 – Modelos dos protetores usados: (a) canaleta protetora (no condutor); (b) cobertura isolante (na cruzeta).



5.3.5.5.2. Isolador

O tipo isolador oferece duas diferenciações dos demais objetos, ele possui uma variável para identificar se ele é o isolador antigo (que está sendo trocado) ou o novo (que será posto no lugar do antigo) e utiliza de animações quando anexado ou desanexado. Ao anexar o isolador ao seu destino, após ser colocado na posição correta inicia-se uma animação dele sendo enroscado ao pino da cruzeta, essa animação afeta somente o isolador, com o ambiente continuando a ser interativo.

O isolador na cena usa o modelo gráfico apresentado na Figura 44 e é tanto um objeto do tipo isolador como um destino de objeto, sendo o destino para o condutor e para o fio de cobre que fixa o condutor ao isolador.

Figura 44 - Modelo usado do isolador.

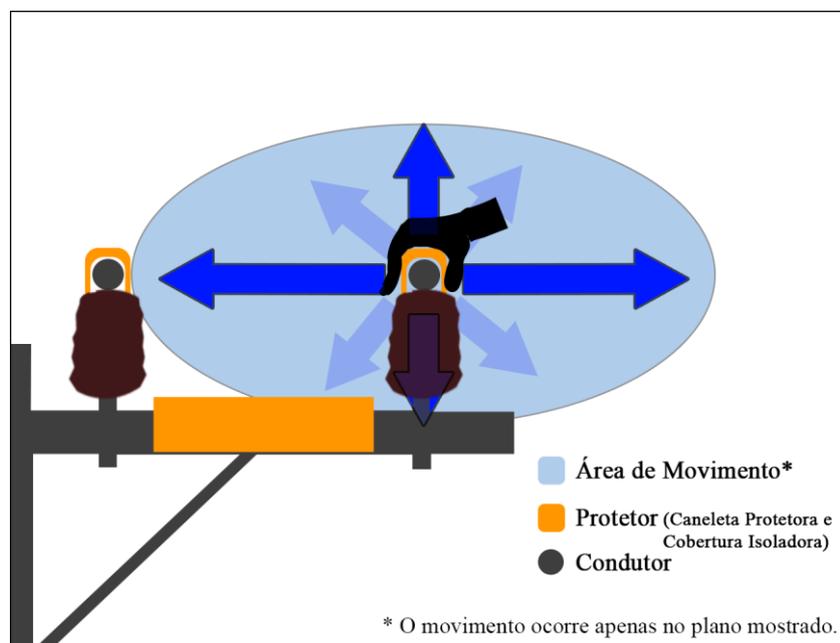


5.3.5.5.3. Condutor

O objeto do tipo condutor utiliza uma movimentação personalizada, pois o condutor no cenário real possui uma movimentação limitada, estando preso nos dois lados em isoladores de outros postes. A movimentação é limitada a valores pré-definidos, onde é utilizando a posição inicial da mão quando o condutor é pego, para calcular a mudança na posição do objeto. Com a posição inicial de interação salva, calcula-se a diferença da posição inicial da mão com a atual, depois filtra esse valor obtido para que ele não ultrapasse os

valores máximos de movimento. Os valores usados para a movimentação do condutor são: no eixo X, 0,5 no eixo Y e 1 no eixo Z, quando o condutor está no isolador, esses valores servem para a modificação nos dois sentidos, como representado na Figura 45. Quando o condutor está na cruzeta a movimentação para trás e para cima possui valores maiores, 2 e 1 respectivamente. Outra limitação no movimento é que o condutor ao ser soltado sempre anexará ao isolador ou a cruzeta, ou seja, sempre anexará ao destino mais próximo.

Figura 45 - Movimentação do condutor a partir do isolador.



5.3.5.6. Ferramenta

As ferramentas são objetos que quando estão sendo segurados permitem interagir com certos objetos que de outra forma não seriam interativos, que serão mostrados na seção 5.3.5.7. As ferramentas possuem uma referência do objeto no qual está sendo usado e armazena uma lista de etiquetas informando em quais objetos ela pode ser usada. A ferramenta possui uma função para verificar as ações realizadas pelo usuário e traduzi-la para uma ação da ferramenta. De forma semelhante aos destinos, o objeto usado com a ferramenta é selecionado quando a mão se aproxima segurando-a, mas o realce é igual aos demais objetos.

5.3.5.6.1. Alicates

O alicate é uma ferramenta utilizada para interagir com o objeto fio de cobre, ela possui duas ações: abrir e fechar. Essas ações quando executadas são repassadas como para o fio de cobre, para que ele execute a ação esperada pelo uso dessa ferramenta. Para auxiliar a interação com o fio de cobre, o alicate não pode ser solto enquanto o fio de cobre está

realçado, ou seja, quando a mão, segurando o alicate, está próxima do fio de cobre. O modelo utilizado para o alicate é apresentado na Figura 46.

Figura 46 - Modelo usado para o alicate



5.3.5.7. Objetos Usados com Ferramenta

Os objetos usados com ferramentas são objetos que só são interativos quando se está segurando a ferramenta adequada. Como base, esses objetos possuem uma referência à ferramenta que está sendo utilizada nele e a declaração da função *AçãoComFerramenta* que deve ser sobrescrita por cada objeto derivado, definindo o que deve ser realizado quando dada ação com a ferramenta é passada para o objeto.

5.3.5.7.1. Fio de Cobre

O fio de cobre é um objeto com o qual só se pode interagir quando se está segurando a ferramenta alicate. Quando o usuário está segurando o alicate e executa uma das ações definidas próximo ao fio de cobre, é então analisado que efeito tal ação tem sobre o objeto e executado. Na Tabela 7 são listados os efeitos exercidos no fio de cobre a partir das ações abrir e fechar do alicate. Como pode ser visto na tabela há quatro informações a serem consideradas, se lado esquerdo e direito do fio de cobre estão presos ao condutor, se o fio de cobre está anexado ao isolador e qual parte do fio de cobre o alicate está mais próximo. Há três pontos de interação no fio de cobre, a ponta esquerda, a ponta direita e o centro. As pontas são utilizadas para prender o fio de cobre ao condutor e só são usadas quando o fio de cobre e o condutor estão anexados ao isolador. O centro é usado quando as duas pontas estão soltas ou quando o fio de cobre não está anexado ao isolador.

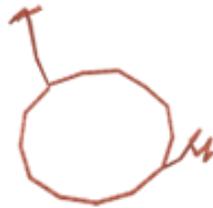
Tabela 7 - Ações executadas no Fio de Cobre.

Lado Esquerdo do Fio de Cobre	Lado Direito do Fio de Cobre	Fio de Cobre	Condição	Ação Realizada pelo Usuário	Efeito Resultante
Preso	Preso	Preso	Mais próximo da esquerda	Fechar	Solta o lado esquerdo do fio de cobre
Preso	Preso	Preso	Mais próximo da direita	Fechar	Solta o lado direito do fio

					de cobre
Preso	Solto	Preso	Mais próximo da esquerda	Fechar	Solta o lado esquerdo do fio de cobre
Preso	Solto	Preso	Mais próximo da direita	Fechar	Prende o lado direito do fio de cobre
Solto	Preso	Preso	Mais próximo da esquerda	Fechar	Prende o lado esquerdo do fio de cobre
Solto	Preso	Preso	Mais próximo da direita	Fechar	Solta o lado direito do fio de cobre
Solto	Solto	Preso	Mais próximo da esquerda	Fechar	Prende o lado esquerdo do fio de cobre
Solto	Solto	Preso	Mais próximo da direita	Fechar	Prende o lado direito do fio de cobre
Solto	Solto	Preso	Mais próximo do centro	Fechar	Remove fio de cobre e o segura
--	--	Solto	--	Fechar	Segura fio de cobre
--	--	Solto	--	Abrir	Solta fio de cobre, anexando caso o destino esteja selecionado

O usuário interage com o ponto de interação que está mais próximo do alicate e que pode interagir. As pontas prendem e soltam quando usada a ação “fechar”, realizando a operação com o uso de uma simulação no objeto fio de cobre, onde a ponta é enrolada em torno do condutor, ou desenrolada caso ela estivesse previamente presa. Quando as duas pontas estiverem soltas, e o alicate estiver mais próximo do centro, a ação “fechar” faz com que o usuário pegue o fio de cobre, removendo-o do isolador. Ao realizar “abrir” o usuário solta o fio de cobre, anexando-o ao isolador, caso esteja selecionado, ou apenas soltando o objeto. Quando o fio de cobre se encontra solto, a única interação é a de pegá-lo e soltá-lo. O modelo utilizado para o fio de cobre pode ser visualizado na Figura 47.

Figura 47 - Modelo usado para o fio de cobre.



5.3.5.8. Mãos

Essa é a entidade que mais se distancia da versão anterior, que trabalha usando o movimento do mouse para mover a mão da personagem nas direções horizontal e vertical. Na nova versão procura-se representar todo movimento realizado com as mãos que está no campo de visão do dispositivo, incluindo rotação da mão e movimentos dos dedos. Para a movimentação da mão com o mouse o usuário está limitado a movê-la em apenas duas direções, pois o mouse é um dispositivo de entrada com valores bidimensionais. Uma maneira de realizar esse controle é a utilização da função *MouseLook*, citada na seção 5.3.4, responsável pela rotação da personagem com os valores do eixo X do mouse e pela rotação da câmera com os valores do eixo Y. Com essas duas rotações afetando o braço se obtém a movimentação da mão que está sempre no meio da tela, causando o efeito de que a personagem olha para onde a mão é movimentada.

Já com o uso do controle Leap Motion o processo é realizado de forma mais complexa devido ao maior número de informações que é possível obter a partir do dispositivo. O Leap Motion SDK obtém informações do dispositivo como, posição e rotação da mão, posição e rotação dos dedos e de suas articulações. Com essas informações é possível reproduzir os movimentos realizados pelas mãos do usuário para o modelo tridimensional apresentado na cena, incluindo movimentos dos dedos.

5.3.5.9. Controlador das Mãos

O principal componente contido nos *assets* importados é sem dúvida o *HandController* (Controlador das Mãos), no qual é feito principalmente o gerenciamento das mãos que estão sendo rastreadas. O objeto controlador de mãos é definido com os objetos gráficos e físicos das mãos esquerda e direita que serão usados quando uma mão começar a ser rastreada.

O controlador é responsável pela criação e destruição dos modelos das mãos quando o rastreamento da mão é iniciado ou perdido, armazenando referências para as mãos ativas da cena. Durante o desenvolvimento do simulador foi adicionada ao controlador: a escolha e

realização da seleção de objetos, o acionamento da ação pegar e soltar objetos e a execução das ações de segurar e soltar objeto, além da interação diferenciada ao segurar uma ferramenta. Esses aspectos serão mostrados nas subseções seguintes.

5.3.5.9.1. Mão Gráfica

O componente responsável pelos aspectos gráficos das mãos possuem o modelo tridimensional da mão, que será usado como avatar para representar as mãos rastreadas do usuário. Esse modelo tridimensional deve possuir uma armadura, que é como um esqueleto da vida real, que permite o a realização de poses com o modelo que provocam a deformação necessária no modelo (BLENDER, 2013). Os ossos dessa armadura são utilizados como controle das articulações das mãos, em que os valores obtidos pelo SDK são usados para controlar o esqueleto. Na Figura 48 é apresentado um exemplo da obtenção das informações dos ossos criados pelo SDK a partir do rastreo, mostrando o retorno da rotação em relação ao controlador das mãos de um osso definido por parâmetro.

Figura 48 - Obtenção dos valores da rotação de osso do dedo usando o SDK.

```
// Retorna a rotação do osso dado em relação ao controlador
public Quaternion GetRotacaoOsso(int osso_tipo) {
    Quaternion rotacao_local =
        dedo_.Bone((Bone.BoneType) (osso_tipo)).Basis.Rotation(espelhar_eixo_z);
    return controlador_.transform.rotation * rotacao_local;
}
```

Fonte: Tradução de trecho do código fonte da classe *FingerModel* do *Leap Motion Core Assets* (LEAP MOTION, 2015).

Na Figura 49 vê-se como é feita a atualização dos dedos do modelo gráfico da mão, em que é feito pela alteração da rotação dos ossos de cada dedo da mão pelo valor obtido do rastreo em relação ao controlador. A atualização da mão é realizada de forma semelhante, como apresentado na Figura 50, atualizando o valor da posição e rotação do osso do modelo que representa a palma da mão, com os valores obtidos pelo rastreo modificados para ficarem relativos ao controlador das mãos. Sendo assim, a representação do movimento das mãos rastreadas é feita fazendo a correspondência dos pontos de articulação do modelo com os pontos definidos pelo SDK do controle Leap Motion, usando como ponto de referência o controlador das mãos.

Figura 49 - Atualiza rotação dos ossos do esqueleto do modelo 3D.

```
public override void AtualizaDedo() {
    for (int i = 0; i < ossos.Length; ++i) {
        if (ossos[i] != null)
            ossos[i].rotation = GetRotacaoOsso(i) * Reorientacao();
    }
}
```

Fonte: Tradução de trecho do código fonte da classe *RiggedFinger* do *Leap Motion Core Assets* (LEAP MOTION, 2015).

Figura 50 - Atualiza rotação e posição do modelo 3D da mão.

```
public override void AtualizaMao() {
    if (palma != null) {
        palma.position = GetPosicaoPalma();
        palma.rotation = GetRotacaoPalma() * Reorientacao();
    }

    if (antebraco != null)
        antebraco.rotation = GetRotacaoBraco() * Reorientacao();

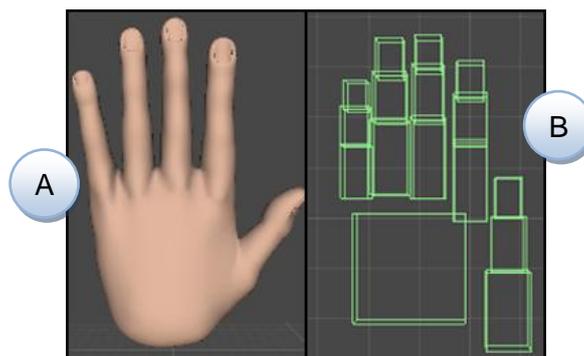
    for (int i = 0; i < dedos.Length; ++i) {
        if (dedos[i] != null)
            dedos[i].AtualizaDedo();
    }
}
```

Fonte: Tradução de trecho do código fonte da classe *RiggedModel* do *Leap Motion Core Assets* (LEAP MOTION, 2015).

5.3.5.9.2. Mão Física

A mão física é o componente responsável pelos aspectos físicos da representação da mão para os cálculos necessários realizados pelo motor de jogos, como colisão. A mão física é composta por blocos retangulares para representar cada osso que composto pelo SDK, esse modelo é usado para simplificar os cálculos físicos, por possuir menos vértices e pode ser vista na Figura 51, que apresenta o modelo gráfico na Figura 51a, e o modelo físico na Figura 51b.

Figura 51 - Modelos de mão usados: (a) modelo gráfico; (b) modelo físico.



Fonte: Captura de Tela dos Modelos Usados no Unity 3D que são Disponibilizados no *Leap Motion Core Assets*.

O modelo físico usado atualiza os seus blocos de modo semelhante ao modelo gráfico, só que ao obter as informações dos ossos do SDK utiliza-os para atualizar a posição e rotação de cada bloco diretamente, em vez de modificar esqueleto de armadura, pois cada bloco é um objeto na cena filho do objeto “mão”. Neste modelo que é armazenada referência ao objeto com o qual está interagindo e que é verificado qual objeto mais próximo da mão, utilizado para decidir qual objeto selecionar e conseqüentemente, interagir.

É também feito no modelo físico o reconhecimento das ações realizadas pelo usuário, que nessa versão é abrir e fechar a mão, para isso é verificado o valor *GrabStrength* (Força de Pegada) fornecido pelo SDK, que informa quão fechada está a mão em uma escala de 0 a 1. Na Figura 52 é apresentada a função responsável por verificar se o usuário está abrindo ou fechando a mão. Primeiro, isso é analisado o valor de *GrabStrength*, considerando a mão fechada caso seja maior que 0.8, isso é feito por um determinado tempo para diminuir a variação de estado resultante de ruído, a partir de testes foi definido o tempo de 0,3 segundo como uma boa alternativa, que diminuía a variação indesejada e não atrasava perceptivamente o reconhecimento das ações.

Figura 52 - Função que verifica ação de Fechar e Abrir.

```
public IEnumerator VerificarFechada(float segundos){
    float tempoPassado = 0.0f;
    int aberto = 0;
    int fechado = 0;
    verificando = true;
    //Verifica qual ação foi realizada
    //(verifica por determinado tempo para diminuir
    //variação indesejada obtida por conta de ruído)
    while(tempoPassado < segundos){
        if(mao_.GrabStrength > forcaPegada){
            fechado++;
        }
        else
            aberto++;
        tempoPassado += Time.deltaTime;
        yield return null;
    }
    //Se a mão está fechada
    if(fechado > aberto){
        //Se já estava fechada
        if(fechado)
            fechando = false;
        //Caso não, atualiza estados e chama
        //função para pegar objeto que está ao
        //alcance
        else{
            fechado = true;
            fechando = true;
        }
    }
    //Continuação
    if(aoAlcance){
        aoAlcance.AoPegar(this);
    }
}
//Se a mão estiver aberta
else{
    //Se antes estava fechada e estiver
    //segurando objeto, chama função para
    //soltá-lo
    if(fechado){
        fechado = false;
        fechando = false;
        if(objetoAtivo){
            objetoAtivo.AoSoltar(this);
        }
    }
}
verificando = false;
yield return null;
}
```

Caso a mão estiver fechada, é verificado se ela já estava fechada antes, caso não, significa que o usuário realizou a ação de fechar e caso haja um objeto válido ao alcance, ele é pego a partir da chamada de função *AoPegar* que faz as ações necessárias pra pegar os

diferentes tipos de objetos. Caso a mão estiver aberta e estava previamente fecha, significa que o usuário abriu a mão, com isso, caso algum objeto estava sendo segurado, ele será solto através da chamada da função *AoSoltar*.

Quando o usuário está segurando uma ferramenta invés de realizar essa verificação acima descrita é chamada a verificação especificada na definição da ferramenta, para reconhecer os gestos específicos para ferramenta e traduzi-los para as ações realizadas por ela. Nessa versão do simulador a ferramenta utilizada aceita os mesmos gestos abrir e fechar como comandos de interação, mas foi mantida essa estrutura para permitir a adição de diferentes comandos para diferentes ferramentas futuramente.

5.3.5.9.3. Escolha de Objeto mais Próximo

A escolha do objeto mais próximo é feita para que haja a seleção e realce do objeto com o qual se pode interagir no momento. Essa escolha é realizada pelo controlador das mãos, permitindo a seleção de apenas um objeto por vez, que ao ser selecionado está sujeito aos comandos de interação realizados pelo usuário. São possíveis 3 tipos de seleções: a seleção de objetos interativos para pegá-los, a seleção de destinos quando se está segurando um objeto para aquele destino e a seleção de objetos interativos com ferramenta enquanto se segura uma ferramenta.

Para decidir qual objeto selecionar o controlador primeiro obtém uma lista de objetos válidos mais próximos de cada mão que se encontra em cena. Um objeto válido são aqueles que se encaixam em um dos casos apresentados na Tabela 8. É obtido um objeto válido, quando houver, a partir do modelo físico de cada mão, que indica o objeto mais próximo da mão que está perto o suficiente.

Tabela 8 - Casos de objetos próximos válidos.

Objeto Segurado	Objeto Próximo Válido
Nenhum	Qualquer, com exceção de destinos e interativos com ferramenta.
Ferramenta	Objetos interativos com ferramenta, que possam ser usados com a ferramenta segurada.
Qualquer	Objeto destino que aceite o objeto segurado para ser anexado.

O modelo físico da mão utiliza a função apresentada na Figura 53, que verifica recursivamente se há um objeto próximo, verificando a partir de uma distância inicial fornecida até uma distância máxima, analisando um número máximo de vezes. A função analisa primeiro se a busca é necessária, verificando que se há um objeto sendo segurado que

ele seja uma ferramenta ou um objeto que pode ser anexado a outro. Se for necessária ela inicia a busca pelo objeto chamando a função *RetorneObjetoMaisPróximo* com o valor da distância inicial, que nessa versão é 0.1. Caso não encontre um objeto válido, a função é chamada recursivamente adicionando o valor passado como distância inicial. Desse modo a mesma função é executada quantas vezes foi definido no *numeroDePassos*. Nessa versão do simulador foi definido o número máximo de passos como 6, portanto após seis execuções da função o valor inicial se encontrará igual ao valor máximo, que nesse caso é 2.

Figura 53 - Busca do objeto mais próximo da mão.

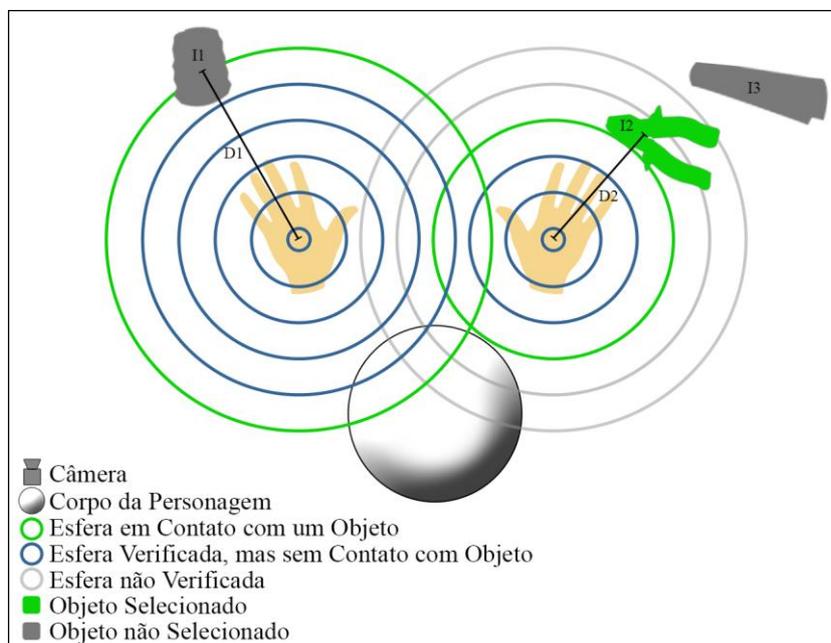
```
public ObjetoBase AcharObjetoMaisProximo(float distanciaVerificacaoInicio,
                                         float distanciaVerificacaoMaxima, float numeroDePassos)
{
    //Caso o objeto segurado não seja do tipo interativo (não possuir destino e não ser ferramenta)
    //encerra busca
    if(objetoAtivo != null && !activeObject.eInterativo()){
        return null;
    }
    else{
        //Verifica se no valor de distância inicial há um objeto válido
        ObjetoBase objetoMaisProximo = RetorneObjetoMaisProximo(distanciaVerificacaoInicio,
                                                                LayerMask.GetMask());

        //Caso não houver, chama a função recursivamente com um valor inicial maior
        if(!objetoMaisProximo)
        {
            if(distanciaVerificacaoInicio <= distanciaVerificacaoMaxima)
            {
                float distanciaPasso = distanciaVerificacaoMaxima / numeroDePassos;
                return AcharObjetoMaisProximo((distanciaVerificacaoInicio + distanciaPasso),
                                             distanciaVerificacaoMaxima, numeroDePassos);
            }
            //Caso não ache nenhum objeto válido até a distância máxima retorna null
            return null;
        }
        //Quando achar, retorna o objeto
        else
        {
            return closestGameObject;
        }
    }
}
```

Como pode ser visto nas Figura 53 e Figura 54, cada mão continua verificando até achar um objeto válido ou até passar da distância máxima que é alcançada após um número fixo de verificações. Quando o valor da distância a ser verificada é definido, é a vez de a função *RetorneObjetoMaisPróximo* analisar se há algum objeto que esteja a essa distância da mão e se ele é válido. Primeiro obtém-se a lista de objetos que colidem com uma esfera de raio igual a distância desejada e que possua o mesmo centro que a mão. Após organizar esses objetos em ordem crescente pelo valor da distância deles até a mão, a lista é percorrida a partir do começo (menor distância) até achar um objeto válido ou encerrar a lista de objetos colididos. Essa verificação é feita analisando se os objetos se encaixam nos casos da Tabela 8

utilizando funções definidas nos objetos que verificam sua interação com os outros objetos, tais como uma função que verifica se o objeto segurado pode ser anexado ao destino e uma função que verifica se a ferramenta segurada pode ser usada no objeto que está sendo analisado.

Figura 54 - Seleção de objeto mais próximo.



Na Figura 54 também é possível observar que dos objetos próximos de cada mão, apenas um dele é selecionado, e essa escolha ocorre no controlador, que tendo a lista de todos os objetos, escolhe o que possui a menor distância entre ele e sua mão correspondente. Após isso, a função de seleção é finalmente chamada.

5.3.5.10. Gerenciadores de Jogo e Cena

Para controlar funções de cena ou gerais do jogo, como mudança de cena, eventos de cena e pausar jogo, foram criados o gerenciador de jogo e os gerenciadores de cena. O gerenciador de jogo é responsável pelo registro do modo de execução, armazenando se a execução sendo feita é sequencial ou não e se é com linha ativa ou desativada. Com essas informações o gerenciador controla a ordem das cenas a serem executadas, realizando a sua transição.

5.3.5.10.1. Gerenciadores de Cena

É definido um gerenciador de cena para cada cena do simulador, a fim de verificar a necessidade de possíveis ações de cena e finalização da cena. Os principais gerenciadores de cena usados são os da cena de preparação e das cenas de manutenção na linha ativa e

desativada. Abaixo são mostradas as responsabilidades desses gerenciadores e uma rápida comparação com o gerenciamento das cenas do ATreVEE 3D.

Na cena de preparação, as duas versões do simulador funcionam da mesma forma, concluindo a cena após pegar todas as ferramentas corretas da cena, a diferença encontrada é que além disso, o gerenciador do ATreVEE IN é responsável pela remoção do objeto da cena, após ele ser pego, invés de ser realizada pelo próprio objeto, como é feito no ATreVEE 3D. O novo gerenciador também passa ao módulo de avaliação os erros e acertos realizados.

As cenas de manutenção possuem gerenciadores muito diferentes dos antigos, os gerenciadores do ATreVEE 3D eram responsáveis pela execução da animação correspondente quando o usuário pressionava algum dos botões gráficos, analisando se os botões foram realizados na ordem certa e encerrando quando o botão “Fim da Operação” é selecionado, como visto no trecho apresentado na Figura 55, em que é carregada a próxima cena após pressionar tão botão.

Figura 55 - Encerramento da cena da troca do primeiro isolador.

```
if (GUI.Button(Rect(Screen.height/2+500,550,Screen.height/4,50),"Fim da Operação")){
    //Passa de fase
    resultadoFinal1 = (varPorcentagem*100)/15;
    PlayerPrefs.SetInt("varPorcentagem",varPorcentagem);
    Application.LoadLevel("CenaFinal01Pontuacao");
}
```

Na Figura 56 pode ser visto o tratamento de um dos botões de ação da interface, o botão responsável pelo isolador, na cena em que é feita a troca do primeiro isolador com a linha ativa. Quando o botão é pressionado pela primeira vez executa a animação de retirar o isolador da cruzeta, quando pressionado pela segunda vez, executa a animação de colocar o isolador e não fazendo mais nada ao ser pressionado outras vezes.

No ATreVEE IN são utilizados dois gerenciadores de cena diferentes, um para a linha ativa e outro para a linha desativada, será utilizado o da linha ativa para a explanação do funcionamento, mas o da linha desativada funciona de forma bem semelhante, removendo os protetores das análises.

O gerenciador de cena de manutenção é informado de toda ação realizada entre os objetos interativos, informando quando o objeto é anexado ou desanexado de outro e quando um objeto é interagido com uma ferramenta. Isso é feito através da junção de estados definidos como enumeradores, cujas declarações são apresentadas na Figura 57, e da referência do objeto com qual foi realizada a ação.

Figura 56 - Gerenciamento do botão gráfico do isolador na troca do primeiro isolador.

```

//Botão de Isolador
if (GUI.Button(Rect(Screen.height/2+500,300,
                    Screen.height/4,100),iconeIsolador)){
    incrementaIsolador++;
    //Caso tenha sido a primeira vez que foi apertado
    if(incrementaIsolador == 1){
        //Inicializa a animação 10, que é retirar o isolador
        animation.Play("acao10");
        //Se antes disso foram realizadas 9 ações corretas
        if(cont == 9){
            //Conta novo acerto
            PlayerPrefs.SetString("acao10","Ok");
            varPorcentagem++;
        }
        //caso não, a ação é considerada errada
        else if(cont != 9){
            PlayerPrefs.SetString("acao10","Erro");
        }
        cont++;
    }
    //Caso tenha sido a primeira vez que foi apertado
    if(incrementaIsolador == 2){
        //Inicializa a animação 11, que é colocar o isolador
        animation.Play("acao11");
        //Se antes disso foram realizadas 10 ações corretas
        if(cont == 10){
            //conta novo acerto
            PlayerPrefs.SetString("acao11","Ok");
            varPorcentagem++;
        }
        else if(cont != 10){
            PlayerPrefs.SetString("acao11","Erro");
        }
        cont++;
    }
}
}

```

Fonte: Projeto do ATreVEE 3D desenvolvido por Santos em (SANTOS, 2013).

Figura 57 - Estados usados na cena de manutenção.

```

public class EstadosJogo : MonoBehaviour {
    public enum Estado{
        Nenhum,
        Protetor_Anexado,
        Protetor_Out,
        IsoladorOld_Anexado,
        IsoladorOld_Desanexado,
        Isolador_Anexado,
        Isolador_Desanexado,
        Fio_Anexado,
        Fio_Desanexado,
        Fio_Prende_Esquerda,
        Fio_Prende_Direita,
        Fio_Solta_Esquerda,
        Fio_Solta_Direita,
        Condutor_Anexado,
        Condutor_Desanexado,
    }
}

```

Esses estados juntamente com a referência do objeto são utilizados para verificar quais passos foram realizados, e com isso identificar quando cada isolador foi trocado,

quando a personagem deve ir para o outro lado do poste para trocar o último isolador e quando a cena está concluída, além de servir também para o módulo de avaliação.

O gerenciador divide essa cena em 3 estágios, um para a troca de cada isolador, através disso são definidos regras para que o estágio seja completado e para realização de ações de cena, quando necessário. Como pode ser visto no trecho de código apresentado na Figura 58 para a conclusão do primeiro estágio é analisado apenas se o primeiro isolador já foi trocado, os demais analisam também se certos protetores foram removidos, antes de continuar. Para concluir o segundo estágio é necessário além de trocar o isolador, retirar os protetores do primeiro fio e da cruzeta, pois ao passar para o próximo estágio a personagem irá ser transferida para o outro lado do poste, impedindo a retirada desses isoladores quando estiver lá. Já o terceiro estágio, é o último da cena que verifica se o terceiro isolador foi trocado e se os protetores restantes foram retirados, para então encerrar a cena. Para a cena com linha desativada, não é feita a verificação dos protetores, sendo necessário apenas o isolador estar trocado.

Figura 58 - Ações de cena do gerenciador de cena de manutenção em linha ativa.

```
public void TrocarEstagio(){
    switch(estagioCorrente){
        case 0: //Estágio 1
            if (isoladorTrocado[0]){ //verifica se o primeiro isolador já foi trocado
                estagioCorrente = 1; //Segundo estágio (o 1o é o 0)
            }
            break;
        case 1: //Estágio 2
            if (isoladorTrocado[1]){ //verifica se o segundo isolador já foi trocado
                //Verifica se foi removido os protetores do lado do poste em
                //que está antes de mudar o personagem de lado
                if(!VerificaSeAnexado(IDObjeto.id.ProtetorC1) && !VerificaSeAnexado(IDObjeto.id.ProtetorD1)
                    && !VerificaSeAnexado(IDObjeto.id.ProtetorE1)){
                    /*
                     * Aqui muda o personagem para o outro lado do poste
                     * para começar a troca do terceiro isolador
                     */
                    estagioCorrente = 2; //Estágio 3
                }
            }
            break;
        case 2: //Estágio 3
            if (isoladorTrocado[2]){
                //Verifica se todos os protetores foram retirados, antes de encerrar a cena
                if(!VerificaSeAnexado(IDObjeto.id.ProtetorC2) && !VerificaSeAnexado(IDObjeto.id.ProtetorD2)
                    && !VerificaSeAnexado(IDObjeto.id.ProtetorE2) && !VerificaSeAnexado(IDObjeto.id.ProtetorE3)){
                    GerenciadorJogo.instance.iniciaProximoNivel();
                }
            }
            break;
    }
}
```

A mudança de estado do isolador para informar que ele foi trocado pode ser visto na Figura 59, que apresenta como é verificado se o primeiro isolador foi trocado. Primeiro verifica se o objeto da última ação recebida é o *Fio1*, que é o fio de cobre que é usado no

primeiro isolador. Caso seja o *Fio1* e a ação realizada tenha sido prender um dos lados do fio de cobre, é verificado se os dois lados do fio de cobre estão presos e se o isolador que está na cruzeta é um isolador novo, caso as assertivas sejam verdadeiras, o estado do primeiro isolador é mudado para informar que ele foi trocado.

Figura 59 - Trecho de código que verifica se o primeiro isolador foi trocado.

```
//Verifica o objeto com o qual foi interagido
switch(objetoInteragido.id){
//Se for um fio de cobre
case IDObjeto.id.Fio1:
//Verifica se a ação realizada foi de prender um dos lados do fio de cobre
if(novo_estado == EstadosJogo.Estado.Arame_Prende_Esquerda ||
    novo_estado == EstadosJogo.Estado.Fio_Prende_Direita){
    FioInterativo fio = AcharObjeto(IDObjeto.id.Fio1) as FioInterativo;
//Se a ação for prender um lado do fio de cobre, verifica se os dois lados estão presos
if(fio && fio.ladoE && fio.ladoD){
    IsoladorInterativo isolador = AcharObjeto(IDObjeto.id.Isolador1) as IsoladorInterativo;
//E se o isolador que está posto é um isolador novo
if(isolador && isolador.eNovo ()){
    //Caso seja, marca o isolador como trocado
    isoladorTrocado[0] = true;
}
}
}
break;
/*
* Definir os blocos "case" para os demais arames
*/
}
```

5.3.5.11. Avaliação

Em todas as cenas com pontuação, a de preparação e as de manutenção, o valor obtido é calculado usando a fórmula apresentada na Figura 60, para obter em porcentagem quantas das ações necessárias foram realizadas e quantas das ações realizadas pelo usuário foram corretas, assim, se o usuário não executar todas as ações ou executar ações erradas sua nota é reduzida.

Figura 60 - Cálculo da pontuação.

$$\frac{\text{Número de Acertos}}{\text{Pontuação Erros} + \text{Total Passos}} \times 100$$

Quando uma ação é realizada pelo usuário o módulo de avaliação é informado, adicionando o número de acertos ou a pontuação de erros. É utilizado pontuação de erros em vez de quantidade de erros, para permitir a atribuição de valores maiores para erros mais graves, como o de pegar no condutor em linha ativa sem ter colocado as proteções. A cena de preparação é composta por 8 ferramentas corretas e tentar pegar qualquer ferramenta errada possui a mesma pontuação. Já as cenas de manutenção possuem valores diferentes para

alguns erros como, pegar no condutor sem proteção, que possui um valor maior, e repetir uma ação desnecessariamente, que possui um valor menor que os demais erros.

A pontuação da manutenção é calculada separadamente para cada isolador, gerando três pontuações independentes que depois são usadas para calcular uma pontuação geral para a cena, através da média aritmética delas. Na Tabela 9 são apresentadas as quantidades de passos para a troca de cada isolador nas duas cenas de manutenção. Essa quantidade de passos se trata da resolução ótima do procedimento que foi apresentada na Figura 19 na página 48, que mostra a sequência de passos para a troca do isolador na linha ativa, e destacando os passos que podem ser removidos para se obter a sequência para a linha desativada. Ao ver os possíveis caminhos apresentados na Figura 19 é possível ver que com 15 passos a ação é concluída para a linha ativa e com 10 passos para a linha desativada.

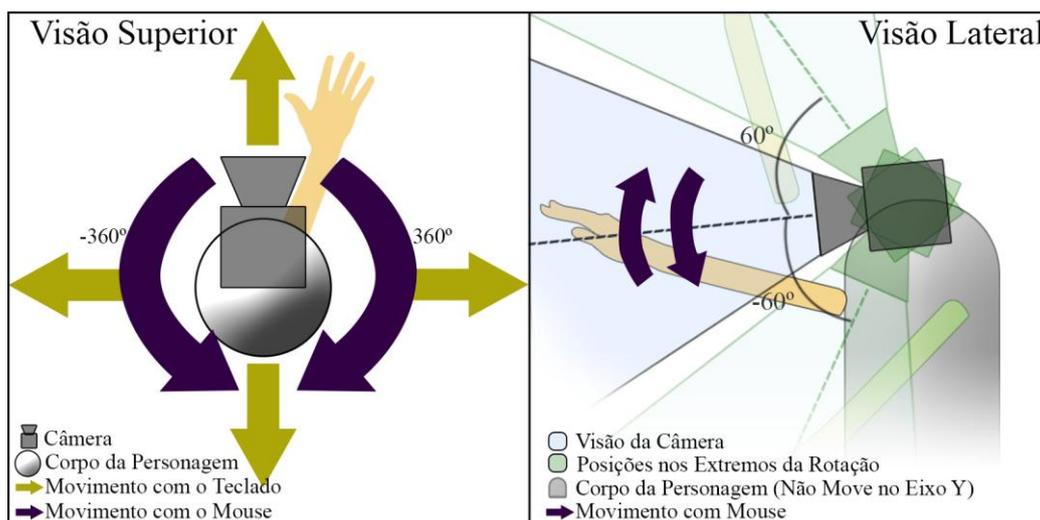
Tabela 9 - Quantidade de passos para cada troca de isolador nas linhas ativa e desativada.

	Quantidade de Passos com a Linha Ativa	Quantidade de Passos com a Linha Desativada
Troca do Primeiro Isolador	15	10
Troca do Segundo Isolador	11	8
Troca do Terceiro Isolador	17	9

5.4. Interação e Controle Leap Motion

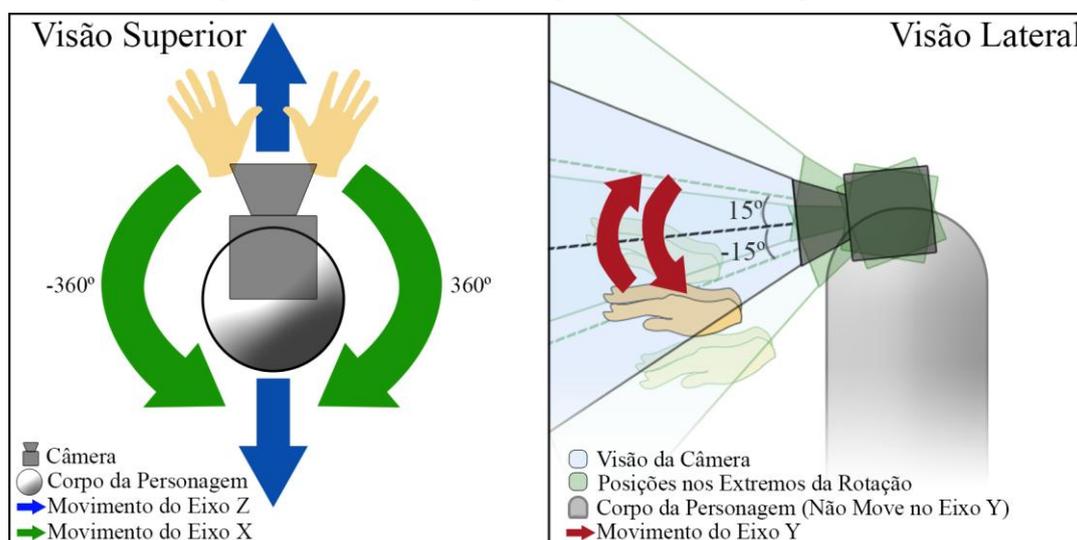
Apesar dos comandos usados e a interpretação destes serem diferentes, como mostrado anteriormente, o controle resultante da personagem é semelhante, como pode ser visto na Figura 61 e na Figura 62. O controle no ATreVEE 3D resulta nos movimentos apresentados na Figura 61, usando o teclado para movimentar a personagem e o mouse para rotacioná-la.

Figura 61 - Movimento do personagem com o mouse.



O controle da personagem com o Leap Motion é realizado de forma semelhante ao mouse e teclado, a maior diferença é como é feito o tratamento dos dados de entrada. Apesar de ser semelhante à versão com mouse, algumas alterações foram realizadas para obter um melhor controle da personagem, elas podem ser vistas na Figura 62. A principal mudança é que a personagem só se movimenta para frente e para trás (relativa a sua rotação), pois a fonte de dados de entrada é uma só, diferente da outra versão que possuía o teclado para movimento e o mouse para rotação. Outra mudança realizada foi a limitação da rotação da câmera, que era de 60° nos dois sentidos e foi alterada para 15°, valor escolhido após alguns testes e considerado ideal para a navegação. Para o cenário de manutenção, mais limitações foram impostas ao movimento, pois o personagem se encontra em uma área de movimentação restrita, e limitando a rotação para as laterais, já que toda a interação é realizada à frente do usuário.

Figura 62 - Movimento da personagem com o controle Leap Motion.



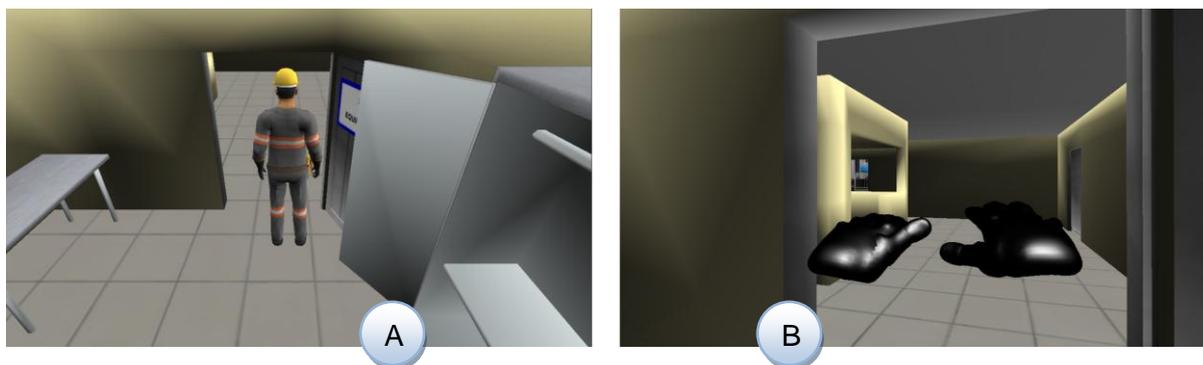
5.5. Cenários

O ATreVEE IN é composto por 5 cenários: Dois Cenários de Transição, um de Preparação e dois de Manutenção, que serão descritos nessa seção. O simulador permite a realização desses cenários pelo Modo Manual, no qual o usuário pode escolher qual cenário deseja realizar e realizá-lo separadamente, e pelo Modo Sequencial, em que são realizados em uma sequência específica.

5.5.1. Cenários de Transição

Os Cenários de Transição são cenários cujo objetivo é ir ao local onde será executada a próxima atividade. Esses cenários só são encontrados no Modo Sequencial, antes de cada cenário e é o ponto de partida desse modo, no qual a personagem se encontra em frente ao prédio onde está a sala de ferramenta que o usuário tem que entrar para iniciar a cena de preparação. O segundo cenário de transição, mostrado na Figura 63, é realizado antes de uma das cenas de manutenção, onde o usuário deve sair do prédio em direção à rua.

Figura 63 - Cenário de transição para o cenário de manutenção: (a) interação com teclado e mouse; (b) interação com o controle Leap Motion.



5.5.2. Cenário de Preparação

O Cenário de Preparação é o cenário que compõe a aplicação piloto, onde o usuário deve escolher e pegar, dentro da sala de ferramentas, os equipamentos e ferramentas que serão utilizados no procedimento de manutenção. Ao finalizar a coleta das ferramentas corretas, é então calculada uma pontuação e mostrada ao usuário, como visto na Figura 65, mostrando na tela as tentativas de pegar ferramentas erradas. A nova versão do cenário de preparação pode ser vista na Figura 64.

Figura 64 - Cenário de preparação.

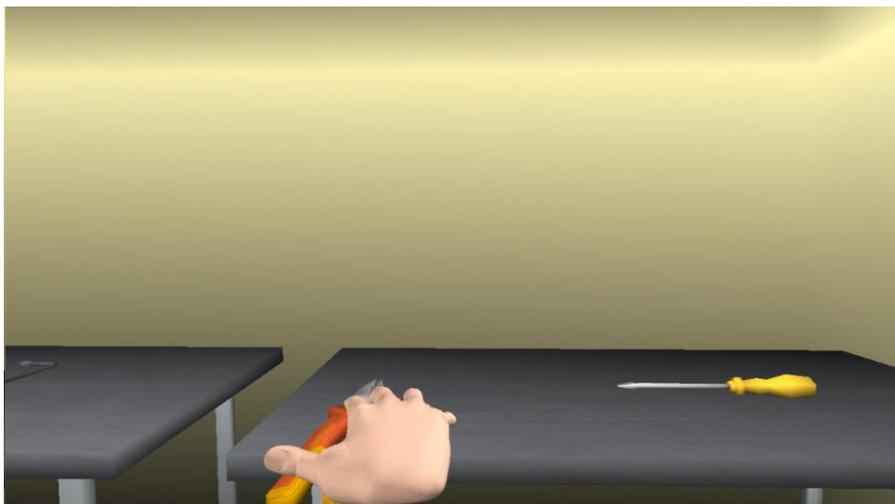


Figura 65 - Tela de pontuação

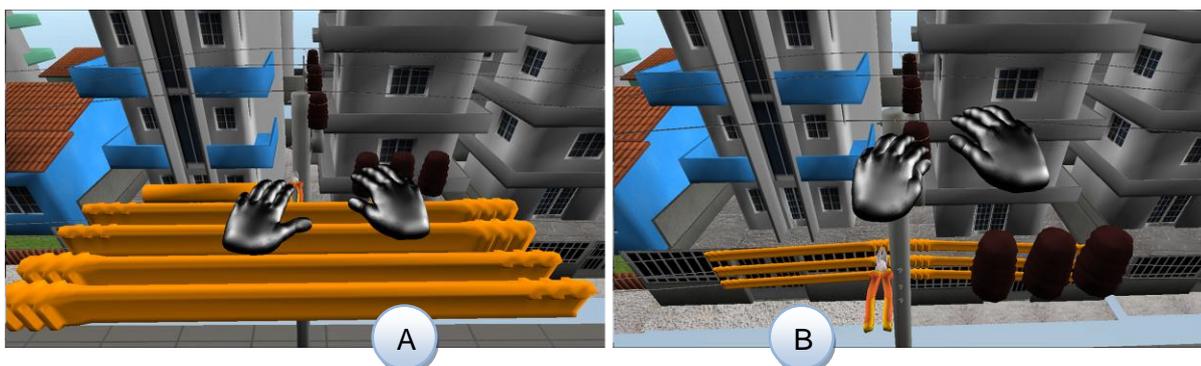


5.5.3. Cenários de Manutenção

Os Cenários de Manutenção são os cenários onde o procedimento é realmente executado, há dois cenários para a realização do procedimento de troca de isoladores, um cenário trata da troca com a linha ativa e ou outro com a linha desativada. Antes da realização do Modo Sequencial é necessária a escolha de em qual das duas situações deseja-se realizar o procedimento, e no Modo Manual, pode ser escolhido qualquer um desses dois cenários para a execução. O procedimento é composto pelos mesmos passos da antiga versão do simulador, mas permite maior liberdade, pois permite a realização dos passos na ordem que o usuário desejar ou achar correta onde, em alguns casos não são reduzidos pontos do usuário, já que alguns passos podem ser realizados em ordens diferentes corretamente, como a escolha de qual lado por o canaleta protetora primeiro.

O cenário possui o conjunto de objetos a serem utilizados alinhados para a escolha do usuário como mostrado na Figura 66. O procedimento de troca é realizado de modo semelhante nas duas situações, sendo que não há a necessidade do uso dos canaletas de proteção com a linha desativada, portanto não são utilizados nesse caso como pode ser visto comparando as imagens do cenário em linha ativa, apresentado na Figura 66a, que possui os isoladores para serem usados, e do cenário em linha desativada, mostrado na Figura 66b, que não os possui.

Figura 66 - Cenários de Manutenção: (a) Linha ativa; (b) Linha desativada.



Diferente do cenário de preparação, na tela de pontuação, que pode ser vista na Figura 67, é mostrada a pontuação individual para cada isolador e a total, e como na preparação, há também um *log* de erros realizados pelo usuário.

Figura 67 - Pontuação do Cenário de Manutenção.



6

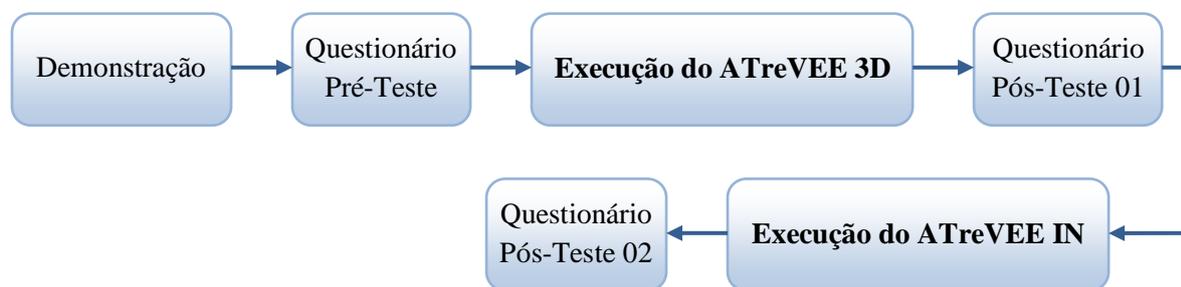
Testes com Usuários e Análise dos Resultados

Neste capítulo são mostrados os testes realizados com voluntários para avaliar a experiência de usuários ao usar com as duas versões do simulador, ATreVEE 3D e ATreVEE IN, usando os resultados obtidos para fazer uma análise comparativa entre as duas versões.

6.1. Método

O método utilizado é composto por 6 etapas, como mostrado na Figura 68. A primeira etapa é a demonstração do procedimento a ser realizado, com a qual é feita uma explanação do procedimento, para que os usuários possam visualizar e aprender as ações necessárias para a realização da tarefa. Após a apresentação das ações a serem executadas, na etapa seguinte, é realizado o preenchimento de um formulário pré-teste, usado para levantamento do perfil dos usuários participantes do experimento. As etapas três e cinco são compostas pela execução do ATreVEE 3D e ATreVEE IN, respectivamente, seguidas pela aplicação de questionários para avaliação da experiência dos usuários.

Figura 68 - Etapas dos testes com usuários.



6.2. Material

Para a execução da primeira etapa do experimento, foi montado um cenário físico para a representação do procedimento. O cenário, que é mostrado na Figura 69, foi construído para representar o poste trifásico e foi composto por uma cruzeta, 3 isoladores, 3 fios de

cobre e 3 tubos metálicos, como representações dos condutores. E para a realização da tarefa, foram usadas as ferramentas necessárias para o procedimento, tais como alicate, cobertura isolante e canaletas protetoras.

Figura 69 - Cenário usado para a Etapa 1.



Os testes foram realizados com as duas versões do simulador: o ATreVEE 3D, apresentado por Santos (2013), que usa os dispositivos mouse e teclado como meios de interação para a realização dos procedimentos; e o ATreVEE IN, apresentado neste trabalho, que utiliza o controlador Leap Motion para interagir com o ambiente virtual e executar as operações desejadas, controlador que conta com 3 LEDs infravermelhos e 2 sensores para detectar as mãos e dedos do usuário e seus movimentos. Por ser o maior foco do treinamento e pelo grande tempo necessário para a realização de todo o teste foi utilizada apenas a cena de manutenção na linha ativa em ambas as versões. Para a execução do ATreVEE 3D foi usado um laboratório com 16 computadores, foi possível o uso de mais de uma máquina na execução do teste já que o simulador necessita apenas dos dispositivos mouse e teclado como meios de entrada. Já o ATreVEE IN, por se disponibilizar de apenas um dispositivo controlador Leap Motion, foi executado em um único computador um usuário por vez.

Para a coleta da opinião dos voluntários foram desenvolvidos questionários para serem preenchidos nas diferentes etapas dos testes. Primeiramente foi construído um questionário pré-teste, que pode ser visto no Anexo A, com o intuito de levantar o perfil de cada participante, procurando informações tais como: faixa etária, afinidade com tecnologia, frequência de uso de jogos eletrônicos e experiência com diferentes meios de interação. Como exemplo das questões contidas no questionário, pode ser usada a seguinte: “Em relação aos novos tipos de controle para videogames, quais você já usou? (Pode marcar mais de um)”, que fornece as opções: “Joystick sem Fio”, “PlayStation Move”, “Wiimote”, “Microsoft Kinect” e “Outros Acessórios do Nintendo Wii”; buscando saber se o usuário possui experiência com meios de dispositivos de interação diferentes que são disponibilizados pelas plataformas de jogos eletrônicos.

Os questionários pós-teste, como podem ser vistos nos Anexo B e Anexo C, são compostos por 23 perguntas, sendo a última uma área para observações adicionais, caso o participante deseje. Na definição das questões foram usados como base, trabalhos como (BURIOL, 2011) e (VORA, NAIR, *et al.*, 2002). As primeiras 22 questões são afirmações, onde o usuário deve marcar quanto ele concorda com cada uma usando escala tipo Likert, indo de “Discordo”, com a nota 1, até “Concordo Fortemente”, com a nota 5. Uma escala tipo Likert é baseada na escala formulada pelo psicólogo Rensis Likert que é composta de pontos fixos estipulados em uma linha, em um sistema de cinco categorias de resposta (pontos), partindo de “aprovo fortemente” até “desaprovo fortemente” (DALMORO e VIEIRA, 2013). Observando os resultados obtidos no trabalho apresentado por Dalmoro e Vieira(2013), que analisa escalas tipo Likert em questionários com diferentes números de itens e diferente disposição das respostas, mostrando resultados em favor da escala tipo Likert de 5 pontos, portanto esse foi o número de itens escolhidos para os questionários aplicados. Procurou-se questões cujas respostas trouxessem informações tais quais: a facilidade de aprendizado de uso do dispositivo, a atenção necessária para a realização dos comandos necessários, a facilidade de navegação, se o sistema pode auxiliar o treinamento dos procedimentos, a facilidade de adaptação ao sistema em relação a faixa etária e conhecimento tecnológico. Para isso, usou-se um método similar aos questionários de Entre as afirmações usadas estão:

- “Interagir com o ambiente virtual foi simples.”
- “Eu pude sempre me concentrar mais na tarefa do que nos comandos necessários para realizar a tarefa.”
- “Esse tipo de aplicação pode aumentar o meu conhecimento.”
- “Esse tipo de aplicação poderia se integrar bem à forma como o treinamento é tradicionalmente realizado.”
- “Eu me senti envolvido na experiência.”
- “Eu me ajustei facilmente ao dispositivo de controle usado para interagir com o ambiente.”
- “A representação visual parece bastante com a realidade.”
- “A forma de navegação é fácil.”

6.3. Experimento

Com o objetivo de avaliar as duas versões do simulador, o ATreVEE 3D e o ATreVEE IN, e compará-las foram realizados testes com usuários. Os testes foram realizados com 24 usuários, alunos de graduação e pós-graduação de Engenharia Elétrica da Universidade Federal de Campina Grande e foram compostos na execução da troca de um isolador nas duas versões, seguidos de questionários.

Inicialmente, foram apresentados os passos que deveriam ser executados (e a ordem que deveriam ser executados) para a troca do isolador. Essa demonstração, representada na Figura 70, foi realizada usando o cenário apresentado na seção anterior, fazendo uso dos instrumentos necessários, como luva e alicate, para realizar as tarefas. Pelo o tempo necessário para a realização da tarefa, nas etapas de demonstração e execução foi usada como procedimento apenas a troca do primeiro isolador. Ao finalizar a explicação e exposição de cada passo da tarefa, os alunos foram direcionados para um laboratório com 16 computadores para realizar as etapas subsequentes. Ao chegar ao laboratório, foram entregues os questionários pré-teste e requisitado que os preenchessem.

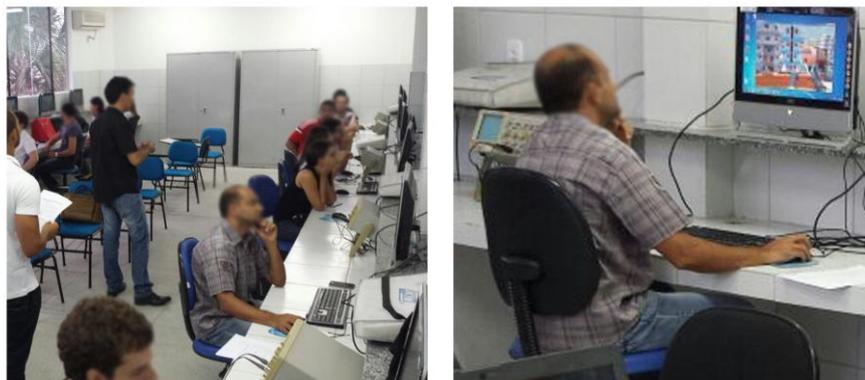
Figura 70 - Demonstração do procedimento para os testes (Etapa 1).



Após a etapa de levantamento de perfil, foi iniciada a explanação do simulador ATreVEE 3D como introdução da terceira etapa dos testes, apresentando a interface gráfica. Seguindo a explicação, foi iniciada a execução do simulador pelos voluntários, mostrada na

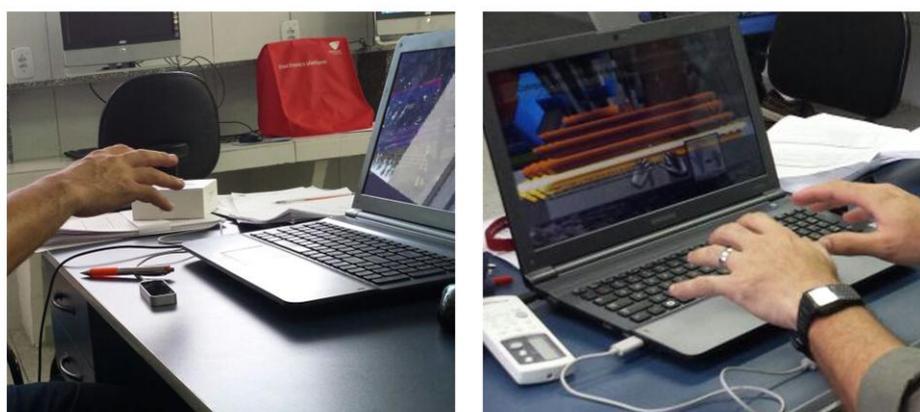
Figura 71. Concluída a execução da simulação foi preenchido o primeiro formulário pós-teste, que pode ser visto no Anexo B.

Figura 71 - Teste com o ATreVEE 3D (Etapa 3).



A quinta etapa do teste, cuja execução pode ser vista na Figura 72, foi a realização da simulação no ATreVEE IN, utilizando o controle Leap Motion que, por só ter acesso a um dispositivo, a execução dessa etapa foi feita com um usuário por vez. De forma semelhante ao que foi feito na etapa da simulação com mouse, foi feita uma explicação rápida do funcionamento do simulador, do dispositivo de entrada e dos comandos de interação, requisitando em seguida a realização da simulação da troca do primeiro isolador na linha ativa pelo modo manual, cujos passos podem ser vistos na Figura 19. Por último foi realizado o preenchimento do segundo formulário pós-teste, referente ao ATreVEE IN.

Figura 72 - Teste com o ATreVEE IN (Etapa 5).



6.4. Resultados

Esses testes foram realizados para coletar informações que pudessem mostrar o potencial desse tipo de aplicação e auxiliar na melhora do sistema, analisando quais pontos devem ser melhorados. Além de prover uma comparação prévia entre as duas interfaces utilizadas nas diferentes versões, permitindo ver as dificuldades dos usuários em cada uma

delas para que possam ser trabalhados os aspectos necessários. Esses resultados, porém não são conclusivos, servem como base para o desenvolvimento de novas etapas do sistema, pois para a análise do impacto do sistema no treinamento é necessário um estudo mais intenso, com uma análise comparativa do treinamento com o auxílio do simulador e do treinamento usando somente meios convencionais (livros, apostilas, vídeos).

Após a execução dos testes e do registro dos dados obtidos com os questionários, foi observado resultados satisfatórios para ambas as versões do simulador, como podem ser vistos nos Anexo F e Anexo G. Com o intuito de comparar as duas versões, as questões foram divididas em dois grupos: as que estão relacionadas com a ajuda de aprendizagem e as que estão relacionadas com os aspectos do uso dos dispositivos. Para simplificação dos dados mostrados, foram retiradas algumas questões que se mostraram muito próximas em interpretação.

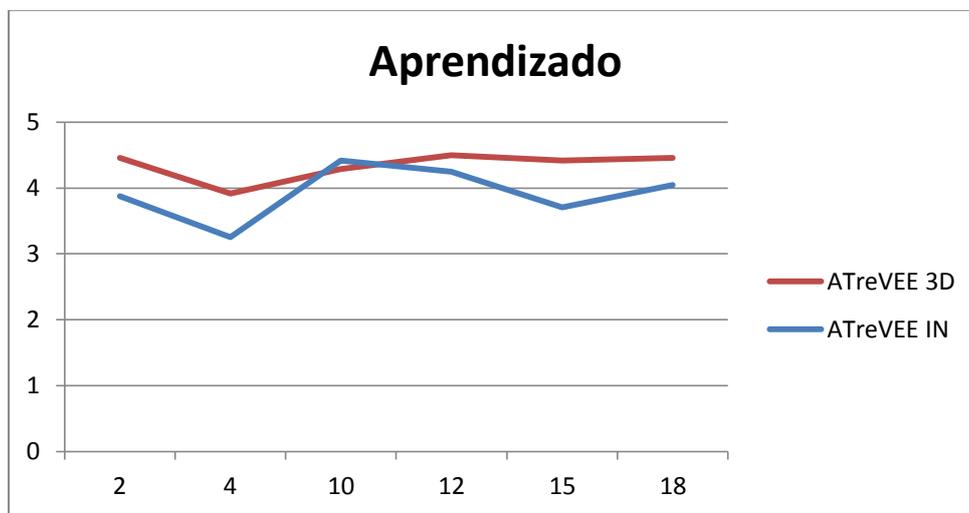
Na Figura 73 são mostrados os resultados relacionados com o quanto os simuladores podem auxiliar a aprendizagem de acordo com os usuários, os resultados foram satisfatórios para ambas as versões. Esses dados mostraram que ao utilizar o ATreVEE 3D, os usuários consideraram ter aprendido a usar o sistema mais rapidamente e notaram uma menor necessidade de concentração nos comandos a serem utilizados, podendo se concentrar mais na tarefa a ser feita. Os usuários também se sentiram mais a vontade e confortáveis ao utilizarem a versão com mouse. Esses pontos são de grande importância e precisam ser trabalhados para simplificar a interação, adicionando intuitividade e proporcionando uma experiência de interação mais transparente, onde o usuário possa focar sua concentração no conteúdo e não na interação. Apesar das dificuldades para se adaptar ao novo meio de interação, os usuários indicaram um maior índice de envolvimento ao usar o ATreVEE IN, já que a interação permitia um maior grau de liberdade, usando gestos e controlando objetos na cena em tempo real em vez de fazer uso apenas de botões, mostrando o potencial do uso da interação gestual em tornar a simulação mais atraente. A lista abaixo contém as perguntas relacionadas ao aprendizado:

2. “Foi rápido aprender como interagir com o ambiente virtual.”
4. “Eu pude sempre me concentrar mais na tarefa do que nos comandos necessários para realizar a tarefa.”
10. “Eu me senti envolvido na experiência.”
12. “Eu gostaria de utilizar com mais frequência esse tipo de aplicação.”

15. “Eu me senti confortável com a simulação.”

18. “Eu me senti tranquilo e à vontade ao utilizar o sistema.”

Figura 73 - Resultados das questões relacionadas com o aprendizado.



Analisando os dados colhidos sobre a utilização dos dispositivos e mostrados na Figura 74, pode-se observar que de acordo com os participantes o uso do mouse se mostrou mais simples e mais fácil de ajustar-se, considerando que sua utilização tornou a navegação mais fácil e beneficiou a interação com o sistema. Esses dados são compreensíveis, já que se trata de um dispositivo que os usuários já estão acostumados e proficientes com seu uso. Mas apesar dos participantes terem escolhido o mouse como o dispositivo mais adequado para a aplicação, eles indicaram o controle Leap Motion como o dispositivo com maior potencial para melhorar o uso do simulador, e observaram uma melhora na habilidade de observar o ambiente virtual e identificar seus objetos, que os usuários classificaram com mais semelhantes aos reais, acredita-se que por conta da maior liberdade de navegação 3D com o uso do Leap Motion no ATreVEE IN, já que foram utilizados os mesmos modelos da versão anterior. As perguntas relacionadas ao uso do dispositivo são listadas abaixo:

1. “Interagir com o ambiente virtual foi simples.”

7. “Eu pude observar e identificar perfeitamente os objetos no ambiente.”

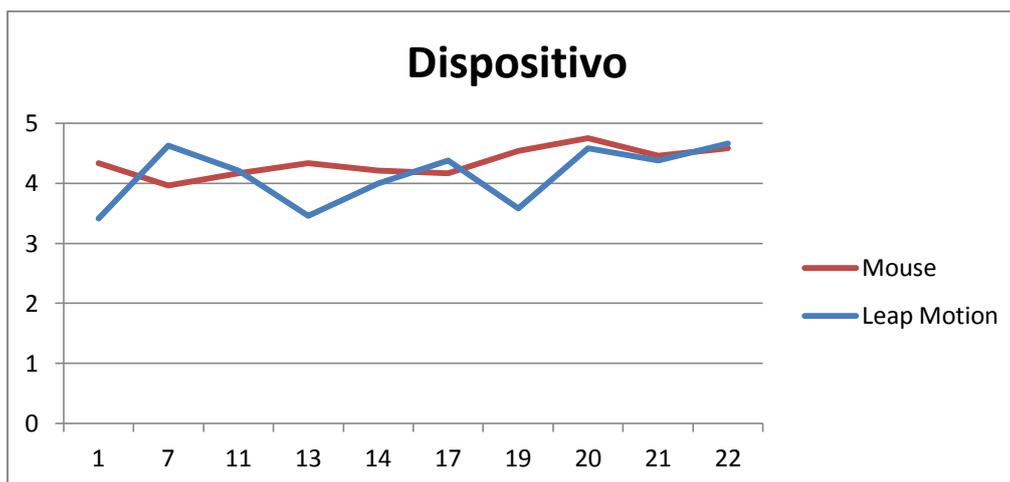
11. “Eu gostei dos gráficos da aplicação.”

13. “Eu me ajustei facilmente ao dispositivo de controle usado para interagir com o ambiente.”

14. “Eu achei muito adequado o dispositivo de interação escolhido para essa aplicação.”

17. “A representação visual parece bastante com a realidade.”
19. “A forma de navegação é fácil.”
20. “Acho que ainda vou lembrar-me de como usar o sistema, mesmo daqui a uma semana.”
21. “Eu achei que o uso do dispositivo na aplicação possibilita melhora da interação.”
22. “Você achou que o uso desse tipo de dispositivo tem a capacidade de melhorar o uso da aplicação?”

Figura 74 - Resultados das questões relacionadas com os dispositivos.

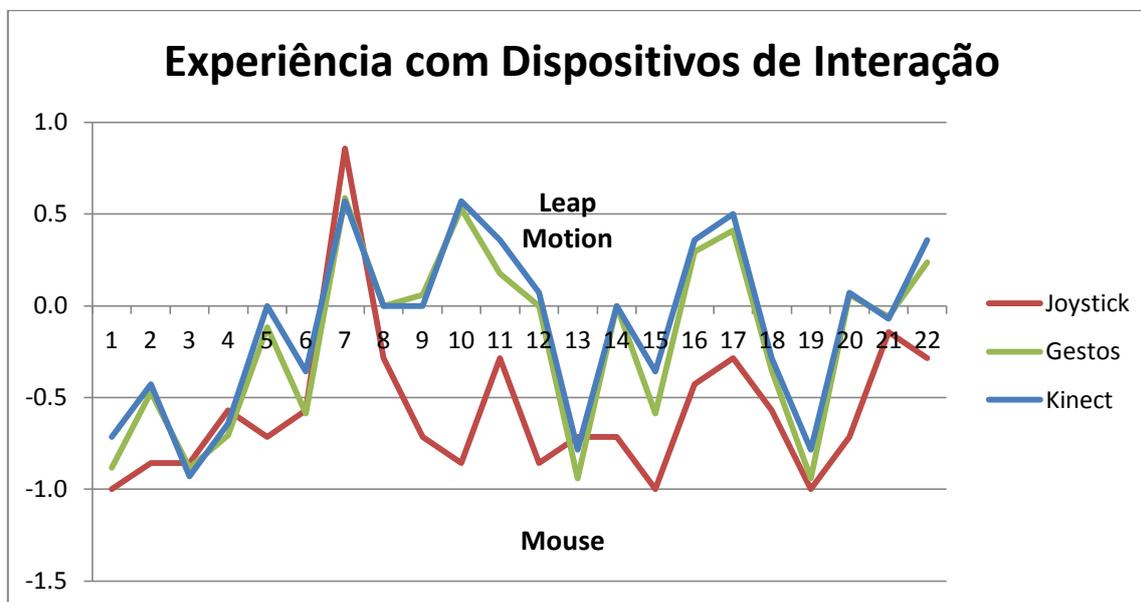


Olhando para todo o conjunto de dados coletados algumas considerações podem ser feitas: os resultados mostram que a adaptação para o novo dispositivo pode ter afetado os resultados, tornando mais difícil para o usuário se concentrar na tarefa enquanto tentavam aprender a usar o dispositivo. Mas o novo dispositivo provocou uma experiência mais envolvente para o usuário, fornecendo uma simulação que pode representar a operação de forma mais realista e com maior liberdade, permitindo uma maior interação dentro do ambiente, controlando os objetos da cena para a realização das tarefas, o que pode trazer melhorias para auxiliar o treinamento.

Essa dificuldade de adaptação pode ser identificada ao observar os resultados colhidos dos usuários que informaram já ter utilizado dispositivos de captura de gestos e comparados com os que nunca usaram. Utilizando as informações fornecidas no formulário de pré-teste, que podem ser vistas no Anexo E, em que o usuário foi questionado com quais dispositivos de interação para sistema de jogos os usuários já haviam tido experiência, foi observado, como pode ser visto na Figura 75, melhores resultados pelos usuários com experiência prévia com dispositivos de interação gestual como, PlayStation Move, Wii Remote ou Microsoft

Kinect. Quando feita a média dos que já haviam utilizado o Kinect que, diferente dos controles PlayStation Move e Wii Remote e semelhante ao controle Leap Motion, não necessita do contato com nenhum dispositivo físico, pode-se observar uma melhora ainda maior.

Figura 75 - Comparação de resultados de usuários com e sem experiência com interação gestual.



Além da falta de adaptação ao uso de gestos para interagir com sistemas computacionais, durante a realização dos testes foi observada outra dificuldade dos usuários no uso do novo dispositivo. Por ser um dispositivo de curto alcance por ser focado no uso das mãos, como mencionado anteriormente, o controle Leap Motion possui uma área de interação pequena, assim ocorre das mãos do usuário escaparem dessa área, o que acaba interrompendo o rastreamento delas e consequentemente, a interação. Apesar de conseguirem se acostumar com o espaço de interação, isso se mostrou um problema para a adaptabilidade dos voluntários, que não possuíam uma representação visual da área na qual deviam manter suas mãos para realizar as tarefas. Por isso é interessante a realização de experimentos com diferentes dispositivos, que possam se adequar melhor ao sistema.

Para melhorar a aceitação e adaptação do novo meio de interação, uma solução pode ser a adição de uma cena para ensinar ao usuário como usar o novo dispositivo e que permita treinar o uso dos comandos, isso pode ser feito com um cenário que instrua o usuário a realizar ações simples, como carregar um objeto de um ponto a outro, explicando quais comandos devem ser feitos e como realizá-los. Outro aspecto a ser estudado é os gestos utilizados para a interação, procurando gestos mais intuitivos e mais fáceis de lembrar, um modo de atingir tal meta é através do uso de um processo que auxilie o desenvolvimento dos

gestos como pode ser visto em (MEDEIROS, TAVARES e FONSECA, 2015). Além desses aspectos, na área de sugestões dos questionários os usuários sugeriram a adição de áudio na simulação, que só está presente na cena de preparação para indicar os erros, e alguns reclamaram da instabilidade do rastreamento, algo que deve ser trabalhado para futuras versões, com a utilização de outros dispositivos ou o melhor uso dos dados captados pelo Leap Motion.

7

Considerações Finais

Nesse trabalho foi projetada e executada a integração de interação natural em simuladores de procedimentos para o setor elétrico e realizado testes com voluntários do setor. Foi utilizado como estudo de caso o simulador de procedimentos ATreVEE 3D, onde foi utilizado o controle Leap Motion como dispositivo para a captação dos movimentos das mãos, que são utilizados para a interação no novo simulador, o ATreVEE IN. Após a execução da integração do simulador com o novo meio de interação foram realizados testes com alunos de Engenharia Elétrica da Universidade Federal de Campina Grande.

Nesse capítulo são apresentadas as conclusões do trabalho na Seção 7.1, as contribuições resultantes da pesquisa na Seção 7.2 e sugestões para trabalhos futuros na Seção 7.3.

7.1. Conclusão

O uso de simuladores virtuais para o treinamento pode trazer grande auxílio para o setor elétrico, pois este é composto por diversas tarefas e procedimentos de alto risco, portanto o treinamento de seus profissionais é de grande importância. Apesar do simulador não substituir o treinamento tradicional e prático, ele auxilia esses treinamentos disponibilizando ao aluno informações extras e a possibilidade de executar os procedimentos diversas vezes, o que seria possível somente em um ambiente físico, seja pelo custo de construir áreas que representem o equipamento real e até mesmo limitação de espaço. Com o simulador há a possibilidade de fácil duplicação do treinamento, permitindo a execução simultânea por mais de um aluno.

Em busca de aprimorar o treinamento via simuladores no setor elétrico esse trabalho realizou o projeto e execução de integração de IN no treinamento do setor elétrico. E para isso, foi realizada uma análise dos trabalhos disponíveis na literatura, buscando as abordagens usadas no setor elétrico e exemplos de boas aplicações de IN em treinamento para indicar o

potencial de melhora que a IN pode gerar. Ainda em busca de analisar a viabilidade do uso dessa interação, foi realizado o desenvolvimento de uma aplicação piloto, com um cenário simplificado, usando o novo meio de interação para investigar as possibilidades de benefícios com a nova interação usada.

Observando o potencial apresentado pela aplicação piloto, foi iniciado o projeto do sistema de treinamento com o uso da interação gestual. Para isso foram levantados os requisitos do sistema, analisando as tarefas que deviam ser cumpridas para definir as funcionalidades necessárias para tal. Após a definição da arquitetura e diagramas, foi então iniciado o desenvolvimento do sistema de treinamento ATreVEE IN, usando as tarefas apresentadas no simulador ATreVEE 3D como referência. O ATreVEE 3D, é um simulador que busca auxiliar o treinamento de operários do setor elétrico, possui atualmente como cenário, a troca de isoladores em postes trifásicos em linhas de 13.8 kV, e sua primeira versão utiliza mouse e teclado como meios de interação.

O ATreVEE IN, desenvolvido nesse trabalho, utiliza o controle Leap Motion para captar os movimentos das mãos do usuário e utilizá-los para interagir com o ambiente virtual, acreditando que o uso de interação natural melhore a simulação, tornando-a mais realista, proporcionando assim um melhor treinamento. O novo simulador busca uma interação com maior grau de liberdade, procurando evitar a limitação do uso de botões nos dispositivos de entrada e na interface gráfica para realizar as tarefas simuladas. Em busca de analisar o potencial inicial de melhora do uso de IN em relação ao uso do simulador ATreVEE 3D, foram realizados testes com ambos os simuladores. Através de testes com estudantes de engenharia elétrica, foi observado que ambas as versões são promissoras no auxílio ao treinamento de procedimentos no setor, e apesar de o ATreVEE 3D ter se sobressaído em quesitos de facilidade de uso e adaptabilidade, o uso do controle Leap Motion demonstrou um maior envolvimento dos usuários na simulação devido a maior liberdade e maior controle na realização dos procedimentos, realizando as ações em si, em vez de usar botões para ativá-las. Além disso, vendo a melhora no desempenho no ATreVEE IN dos usuários com alguma experiência com interação gestual, conclui-se que a adaptação prévia com os dispositivos de entrada foi de grande influência nesses resultados.

Acredita-se que a adaptabilidade ao mouse na simulação se dê ao uso costumeiro desse dispositivo no dia-a-dia em contraste ao uso de um novo meio de interação, essa crença vem da comparação dos resultados fornecidos pelos usuários que já haviam tido contato com interação gestual e os demais. Portanto com o estudo de uma interface mais simples e com a

adaptação dos usuários a essa nova tecnologia, espera-se melhora aos sistemas de treinamento, além do aumento do grau de liberdade na interação e navegação em ambientes virtuais, já que possibilita a entrada de comandos não possíveis via mouse e teclado, como comandos gestuais e tridimensionais para navegação.

7.2. Contribuições

As contribuições deste trabalho foram:

- Estudo sobre o uso de simuladores de treinamento e de trabalhos que os utilizem no setor elétrico;
- Projeto para o desenvolvimento de simulador de treinamento com o uso de Interação Natural;
- O estudo do simulador de procedimentos no setor elétrico, ATreVEE 3D, para integração de Interação Natural;
- O desenvolvimento do ATreVEE IN, permitindo um novo conjunto de comandos e ações na interação e tornando-a mais próxima da realidade;
- A investigação da aceitação de dispositivos de Interação Natural por estudantes do setor elétrico, através dos testes realizados com o ATreVEE 3D e ATreVEE IN, cujos resultados foram analisados e discutidos para apontar em que aspectos o sistema com IN pode ser aprimorado;
- As seguintes publicações:
 - Full Paper: A Utilização de Interação Natural em Ambientes Tridimensionais para Treinamento no Setor Elétrico. CBIE - Workshop de Realidade Virtual e Aumentada aplicada à Educação, 2014;
 - Short Paper: Ambiente Computacional Baseado em Realidade Virtual e Interação Natural para Treinamento no Setor Elétrico. TISE - Conferência Internacional sobre Informática na Educação, 2014.

7.3. Trabalhos Futuros

O trabalho aqui desenvolvido contou com a realização de testes com usuários para validar o sistema desenvolvido, mas não foram extensivos suficientes para avaliar o impacto desse sistema no treinamento dos profissionais do setor elétrico, como trabalho futuro pode ser feito esse estudo, avaliando o impacto do simulador em relação ao treinamento convencional.

É possível também trabalhar no crescimento e melhora do simulador, com o desenvolvimento de cenários implementando novos procedimentos. Um estudo mais detalhado para melhorar a interface natural também pode ser feito, podendo trabalhar com o uso de novos dispositivos e novos comandos que possam se adequar melhor ao cenário da simulação.

O uso de aspectos de jogos eletrônicos pode ajudar na eficácia do treinamento, pode ser feito um estudo da utilização desses aspectos na simulação, tais como o uso de níveis de dificuldades e a implementação de um sistema de classificação para comparar as pontuações dos usuários, estimulando competição saudável entre os participantes.

Referências

ABATE, A. F. et al. A Haptic-Based Approach to Virtual Training for Aerospace Industry. **Journal of Visual Languages and Computing**, v. 20, n. 5, p. 318–325, 2009.

ANDROID. Android SDK | Android Developers. **Android Developers**, 2015. Disponível em: <<http://developer.android.com/sdk/index.html>>. Acesso em: Maio 2015.

APPLE. Develop apps for iPad - Apple Developer. **Apple Developer**, 2008. Disponível em: <<https://developer.apple.com/ipad/sdk/>>. Acesso em: Novembro 2013.

ARENDARSKI, B.; TERMATH, W.; MECKING, P. Maintenance of Complex Machines in Electric Power Systems Using Virtual Reality Techniques. **2008 IEEE International Symposium on Electrical Insulation (ISEI 2008)**, Vancouver, Canadá, 2008. p. 483-487.

BARATA, P. A. B. Projetos e Implementação de um Sistema de Autoria para Animações, Simulações e Treinamentos em Realidade Virtual. **Dissertação de Mestrado. Programa de Pós-Graduação em Engenharia Elétrica da Universidade do Pará**, Belém, PA, Setembro 2010.

BELLOC, O. R. Um Arcabouço para o Desenvolvimento de Simuladores de Procedimentos. **Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo (USP)**, São Paulo, SP, 2011.

BIANCHINI, R. C. et al. Jogos Eletrônicos e Realidade Virtual. Em: TORI, R.; KIRNER, C.; SISCOOTTO, R. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Livro do Pré-Simpósio VIII Symposium on Virtual Reality (SVR)**. Belém, PA: [s.n.], 2006. Cap. 12, p. 199-219.

BLENDER. Rigging - Introduction. **Blender Reference Manual**, 2013. Disponível em: <<http://www.blender.org/manual/rigging/introduction.html>>. Acesso em: Maio 2015.

BUNKER, A. Exclusive: How does Microsoft Xbox Kinect work? **T3**, 2011. Disponível em: <<http://www.t3.com/features/exclusive-how-does-microsoft-xbox-kinect-work>>. Acesso em: Agosto 2012.

BURIOL, T. M. Convergência de Games e Realidade Virtual para Treinamento de Manutenção em Redes de Energia em Linha Viva. **Defesa de Doutorado. Programa de Pós-Graduação em Métodos Numéricos em Engenharia. Universidade Federal do Paraná**, Curitiba, PR, 2011.

BURIOL, T. M. Convergência de Games e Realidade Virtual para Treinamento de Manutenção em Redes de Energia em Linha Viva. **Defesa de Doutorado. Programa de Pós-Graduação em Métodos Numéricos em Engenharia. Universidade Federal do Paraná**, Curitiba, 2011.

CARDOSO, A. et al. VRCEMIG: a Virtual Reality System for Real Time Control of Electric Substations. **IEEE Virtual Reality 2013**, Orlando, FL, EUA, 2013. p. 165-166.

CHEN, L.; WEI, H.; FERRYMAN, J. A Survey of Human Motion Analysis Using Depth Imagery. **Pattern Recognition Letters**, v. 30, n. 15, p. 1995–2006, 2013.

COGE. Relatório 2012 - Estatísticas de Acidentes do Setor Elétrico Brasileiro. **Fundação COGE**, 2013. Disponível em: <<http://www.funcoge.org.br/csst//relat2012/index.html>>. Acesso em: Fevereiro 2014.

COLGAN, A. How Does the Leap Motion Controller Works? **Blog Leap Motion**, 2014. Disponível em: <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/?utm_source=Leap+Motion+Developer+Newsletter&utm_campaign=2b1386282b-Developer_Newsletter_74&utm_medium=email&utm_term=0_d7eaf93515-2b1386282b-66494525>. Acesso em: Agosto 2014.

CUNHA, I. L. L. et al. Cut and Suture Support on Volumetric Models in the CyberMed Framework. **Procedia Technology. 4th Conference of ENTERprise Information Systems – aligning technology, organizations and people (CENTERIS 2012)**, 2012. p. 771–776.

DALMORO, M.; VIEIRA, K. M. Dilemas na Construção de Escalas Tipo Likert: O Número de Intens e a Disposição Influenciam nos Resultados? **Revista Gestão Organizacional (RGO)**, v. 6, p. 161-174, 2013.

EL-KHALILI, N. H.; BRODLIE, K. W. Surgical Training on the Web. **Future Generation Computer Systems**, v. 17, n. 2, p. 147–158, 2000. ISSN 0167-739X.

FERREIRA, L. F. G.; TAROUCO, L. M. R.; BECKER, F. Fazer e Compreender na Realidade Virtual: em Busca de Alternativas para o Sujeito da Aprendizagem. **RENOTE - Revista Novas Tecnologias na Educação**, Porto Alegre, RS, v. 1, n. 1, p. 1-11, 2003.

FERREIRA, T. V. Estimação Inteligente da Poluição de Isolamentos Elétricos Baseada nos VECSE do Ruído Ultrassônico. **Tese de Doutorado. Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande**, Campina Grande, PB, 2011.

FITZGERALD, M.; RIVA, G. Virtual Reality. Em: BEOLCHI, L. **Telemedicine Glossary**. [S.l.]: European Commission-DG INFSO, 2001. p. 327-329.

GARANT, E. et al. A Virtual Reality Training System for Power-Utility Personnel. **IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing**, Victoria, BC, Canadá, 1995. p. 296-299.

GEOMAGIC. Geomagic Touch Overview. **Sensable**, 2013a. Disponível em: <<http://geomagic.com/en/products/phantom-omni/overview>>. Acesso em: Dezembro 2013.

GEOMAGIC. OpenHaptics Toolkit Overview. **Sensable**, 2013b. Disponível em: <<http://geomagic.com/en/products/open-haptics/overview/>>. Acesso em: Dezembro 2013.

GIL-O-TOPIA. How does the Kinect really work? **Gil-o-topia**, 2010. Disponível em: <<http://gilotopia.blogspot.com.br/2010/11/how-does-kinect-really-work.html>>. Acesso em: Setembro 2012.

GOSSELIN, F. et al. Haptic Systems for Training Sensorimotor Skills: A Use Case in Surgery. **Robotics and Autonomous Systems**, v. 61, n. 4, p. 380-389, 2013.

HANDFORD, D.; ROGERS, A.; CROSS, K. Agent-based Traffic Operator Training Environments for Evacuation Scenarios. **2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)**, Lyon, França, v. 2, 2011. p. 438-441.

HTK. **HTK Speech Recognition Toolkit**. **HTK**, 1995. Disponível em: <<http://htk.eng.cam.ac.uk>>. Acesso em: Maio 2014.

HUERTA, J. et al. Reusable Dialog Components. **IBM**, 2004. Disponível em: <<http://www.ibm.com/developerworks/web/library/wa-dialogcomp/index.html>>. Acesso em: Novembro 2012.

KIRNER, C.; KIRNER, T. G. Evolução e Tendências da Realidade Virtual e da Realidade Aumentada. Em: RIBEIRO, M. W. S.; ROBERTO, E. **Realidade Virtual e Aumentada: Aplicações e Tendências**. XIII Symposium on Virtual and Augmented Reality (SVR). Livro do Pré-simpósio. Uberlândia, MG, Brasil: SBC, 2011. Cap. 1, p. 10-25.

LEAP MOTION. Leap Motion. **Leap Motion**, 2013. Disponível em: <www.leapmotion.com/>. Acesso em: Dezembro 2013.

LEAP MOTION. C# SDK Documentation - Leap Motion C# SDK v2.2 documentation. **Leap Motion Developer Portal**, 2014. Disponível em: <<https://developer.leapmotion.com/documentation/csharp/index.html>>. Acesso em: Abril 2015.

LEAP MOTION. Leap Motion Core Assets - Asset Store. **Unity 3D Asset Store**, 2015. Disponível em: <<https://www.assetstore.unity3d.com/en/#!/content/17770>>. Acesso em: Maio 2015.

LOS ARCOS, J. L. et al. PDA: A Training Application to Electrical Substation Operation. **IEEE International Conference on Multimedia Computing and Systems**, Florença, Itália, v. 2, 1999. p. 1058-1060.

MANTOVANI, F. VR Learning: Potential and Challenges for the Use of 3D Environments in Education and Training. Em: RIVA, G.; GALIMBERTI, C. **Towards Cyber-Psychology: Mind, Cognitions and Society in the Internet Age**. p. 207-226. ed. Amsterdã, Países Baixos: IOS Press, 2001. Cap. 12.

MEDEIROS, A. C. S.; TAVARES, T. A.; FONSECA, I. E. D. **How to Design an User Interface Based on Gestures**. 17th International Conference on Human-Computer Interaction (HCI International 2015). Los Angeles, CA, EUA: [s.n.]. 2015.

MICROSOFT. Microsoft Speech API (SAPI) 5.3. **MSDN**, 2009. Disponível em: <<http://msdn.microsoft.com/en-us/library/ms723627.aspx>>. Acesso em: Novembro 2013.

MICROSOFT. Microsoft Surface 2.0 SDK. **MSDN**, 2011. Disponível em: <[http://msdn.microsoft.com/en-us/library/ff727815\(Surface.20\).aspx](http://msdn.microsoft.com/en-us/library/ff727815(Surface.20).aspx)>. Acesso em: Novembro 2013.

MILES, H. C. et al. A Review of Virtual Environments for Training in Ball Sports. **Computers & Graphics**, v. 36, n. 6, p. 714-726, 2012.

PAMPLONA JR., A.; RIBEIRO FILHO, M.; MOREIRA, C. Avaliação do Sistema de Autoria de Instruções Técnicas Virtuais. **XII Symposium on Virtual and Augmented Reality**, Natal, RN, 2010.

PANAIT, L. et al. The Role of Haptic Feedback in Laparoscopic Simulation Training. **Journal of Surgical Research**, v. 156, n. 2, p. 312-316, 2009.

PINHO, M. S. Interação em Ambientes Tridimensionais. **Minicurso para 3º Workshop de Realidade Virtual, SBC**, Porto Alegre, 2000. Disponível em: <http://www.inf.pucrs.br/~pinho/3DInteraction/#_Toc493643444>.

PONDER, M. et al. Immersive VR Decision Training: Telling Interactive Stories Featuring Advanced Virtual Human Simulation Technologies. **Eurographics Workshop on Virtual Environments (EGVE '03)International Immersive Projection Technologies Workshop**, Nova York, NY, EUA, 2003. p. 97-106.

PRIMESENSE. Our Full 3D Sensing Solution. **PrimeSense Natural Interaction**, 2011b. Disponível em: <<http://www.primesense.com/en/component/content/article/9-solutions/115-the-primesense-3d-sensing-solution>>. Acesso em: Agosto 2012.

ROBISON, R. A.; LIU, C. Y.; APUZZO, M. L. J. Man, Mind, and Machine: The Past and Future of Virtual Reality Simulation in Neurologic Surgery. **World Neurosurgery**, v. 76, n. 5, p. 419-430, 2011.

SANTOS, F. G. Plataforma Computacional para Ensino e Treinamento de Profissionais do Setor Elétrico. **Dissertação de Mestrado. Programa de Pós-Graduação em Sistemas de**

Comunicação e Automação na Universidade Federal RURAL do Semi-Árido, Mossoró, RN, 2013.

SEMERARO, F. et al. Motion Detection Technology as a Tool for Cardiopulmonary Resuscitation (CPR) Quality Training: A Randomised Crossover Mannequin Pilot Study. **Resuscitation**, v. 84, 2013. 501– 507.

SOUSA, M. P. A. et al. Maintenance and Assembly Training in a Hydroelectric Unit of Energy Using Virtual Reality Desktop. **IEEE Latin America Transactions**, v. 6, n. 5, Setembro 2008. p. 484-491.

THALMIC LABS. Myo - Gesture control armband by Thalmic Labs. **Thalmic Labs**, 2014. Disponível em: <<https://www.thalmic.com/en/myo/>>. Acesso em: Abril 2014.

TORI, R.; KIRNER, C. Fundamentos de Realidade Virtual. Em: TORI, R.; KIRNER, C.; SISCOOTTO, R. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Livro do Pré-Simpósio VIII Symposium on Virtual Reality (SVR)**. Belém, PA: Ed. SBC, 2006. Cap. 1, p. 2-21.

TURK, M. Multimodal Interaction: A Review. **Pattern Recognition Letters**, v. 36, 2014.

UNITY. Unity - Game Engine. **Unity Technologies**, 2014. Disponível em: <<http://unity3d.com/pt>>. Acesso em: Maio 2014.

VALLI, A. **Notes on Natural Interaction**. [S.l.]: [s.n.], 2007.

VORA, J. et al. Using Virtual Reality Technology for Aircraft Visual Inspection Training: Presence and Comparison Studies. **Applied Ergonomics**, v. 33, n. 2, p. 559-570, 2002.

VOURVOPOULOS, A.; LIAROKAPIS, F. Evaluation of Commercial Brain–computer Interfaces in Real and Virtual World Environment: A Pilot Study. **Computers and Electrical Engineering**, v. 40, n. 2, p. 714-729, 2013.

WASFY, A.; WASFY, T.; NOOR, A. Intelligent Virtual Environment for Process Training. **Advances in Engineering Software**, v. 35, n. 6, p. 559-570, 2004.

WITMER, B. G.; SINGER, M. J. Measuring Presence in Virtual Environments: A Presence Questionnaire. **Presence**, Massachusetts Institute of Technology (MIT), Massachusetts, EUA, v. 7, n. 3, 1998.

YONCK, R. The Age of the Interface. **The World Future Society**, v. 44, n. 2, 2010.

ANEXOS

ANEXO A - Formulário Usado Antes da Execução do Simulador.

Pré-Teste

* **Requerido**

1. Em que faixa etária você se enquadra?*

- 11 a 20 anos
- 21 a 30 anos
- 31 a 40 anos
- 41 a 50 anos
- 51 a 60 anos

2. Em relação à utilização de dispositivos tecnológicos, em qual item você se enquadra?*

- Sei escrever programas
- Considero-me um usuário avançado
- Considero-me um usuário intermediário
- Considero-me um usuário principiante
- Não entendo muito de computadores

3. Em relação a jogos de computador, ou videogame, em qual item você se enquadra?*

- Praticamente nunca joguei
- Jogava no passado mas não jogo mais
- Jogo eventualmente
- Jogo com frequência
- Jogo muito! Sempre que tenho um tempo livre

4. Em relação aos modernos videogames, quais você já usou? (Pode marcar mais de um)*

- XBox 360/ XBox One
- Play Station 3/ Play Station 4
- Nintendo Wii/ Nintendo Wii U
- Jogos de PC
- Outros

5. Em relação aos novos tipos de controle para videogames, quais você já usou? (Pode marcar mais de um)*

- Joystick sem fio
- Play Station Move
- Wiimote
- Microsoft Kinect
- Outros acessórios do Nintendo Wii

6. Se você joga com dispositivos de gestos (Move, Kinect), em qual item você se enquadra?

- Uso de gestos melhora muito o controle do jogo
- Uso de gestos melhora um pouco o controle do jogo
- Uso de gestos não melhora o controle do jogo
- Uso de gestos dificulta um pouco o controle do jogo
- Uso de gestos dificulta muito o controle do jogo

7. O que você sabe sobre Realidade Virtual?*

- Nem sei o que é
- Sei o que é, mas nunca experimentei
- Sei o que é, já experimentei
- Conheço bem, mas nunca cheguei a trabalhar com Realidade Virtual
- Já trabalhei com Realidade Virtual

ANEXO B - Formulário Usado Após a Execução do Simulador com Mouse.**ATreVEE 3D – Versão 1 (Mouse)***** Requerido****1. Interagir com o ambiente virtual foi simples***

1 2 3 4 5

Discordo Concordo Fortemente**2. Foi rápido aprender como interagir com o ambiente virtual.***

1 2 3 4 5

Discordo Concordo Fortemente**3. Eu me senti proficiente em interagir com o ambiente virtual no final da aplicação.***

1 2 3 4 5

Discordo Concordo Fortemente**4. Eu pude sempre me concentrar mais na tarefa do que nos comandos necessários para realizar a tarefa.***

1 2 3 4 5

Discordo Concordo Fortemente**5. As interações com o ambiente parecem naturais.***

1 2 3 4 5

Discordo Concordo Fortemente**6. Os gestos utilizados se assemelham com a ação real.***

1 2 3 4 5

Discordo Concordo Fortemente**7. Eu pude observar e identificar perfeitamente os objetos no ambiente.***

1 2 3 4 5

Discordo Concordo Fortemente**8. Esse tipo de aplicação pode aumentar o meu conhecimento.***

1 2 3 4 5

Discordo Concordo Fortemente

9. Esse tipo de aplicação poderia se integrar bem à forma como o treinamento é tradicionalmente realizado.*

1 2 3 4 5

Discordo Concordo Fortemente

10. Eu me senti envolvido na experiência.*

1 2 3 4 5

Discordo Concordo Fortemente

11. Eu gostei dos gráficos da aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

12. Eu gostaria de utilizar com mais frequência esse tipo de aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

13. Eu me ajustei facilmente ao dispositivo de controle usado para interagir com o ambiente.*

1 2 3 4 5

Discordo Concordo Fortemente

14. Eu achei muito adequado o dispositivo de interação escolhido para essa aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

15. Eu me senti confortável com a simulação.*

1 2 3 4 5

Discordo Concordo Fortemente

16. Eu me senti envolvido com a simulação.*

1 2 3 4 5

Discordo Concordo Fortemente

17. A representação visual parece bastante com a realidade.*

1 2 3 4 5

Discordo Concordo Fortemente

18. Eu me senti tranquilo e à vontade ao utilizar o sistema.*

1 2 3 4 5

Discordo Concordo Fortemente

19. A forma de navegação é fácil .*

1 2 3 4 5

Discordo Concordo Fortemente

20. Acho que ainda vou lembrar-me de como usar o sistema, mesmo daqui a uma semana.*

1 2 3 4 5

Discordo Concordo Fortemente

21. Eu achei que o uso do dispositivo na aplicação possibilita melhora da interação.*

1 2 3 4 5

Discordo Concordo Fortemente

22. Você achou que o uso desse tipo de dispositivo tem a capacidade de melhorar o uso da aplicação?*

1 2 3 4 5

Discordo Concordo Fortemente

23. Alguma Sugestão?

ANEXO C - Formulário Usado Após a Execução do Simulador com o Controle Leap Motion.

ATreVEE 3D – Versão 2 (Leap Motion)

* Required

1. Interagir com o ambiente virtual foi simples*

1 2 3 4 5

Discordo Concordo Fortemente

2. Foi rápido aprender como interagir com o ambiente virtual.*

1 2 3 4 5

Discordo Concordo Fortemente

3. Eu me senti proficiente em interagir com o ambiente virtual no final da aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

4. Eu pude sempre me concentrar mais na tarefa do que nos comandos necessários para realizar a tarefa.*

1 2 3 4 5

Discordo Concordo Fortemente

5. As interações com o ambiente parecem naturais.*

1 2 3 4 5

Discordo Concordo Fortemente

6. Os gestos utilizados se assemelham com a ação real.*

1 2 3 4 5

Discordo Concordo Fortemente

7. Eu pude observar e identificar perfeitamente os objetos no ambiente.*

1 2 3 4 5

Discordo Concordo Fortemente

8. Esse tipo de aplicação pode aumentar o meu conhecimento.*

1 2 3 4 5

Discordo Concordo Fortemente

9. Esse tipo de aplicação poderia se integrar bem à forma como o treinamento é tradicionalmente realizado.*

1 2 3 4 5

Discordo Concordo Fortemente

10. Eu me senti envolvido na experiência.*

1 2 3 4 5

Discordo Concordo Fortemente

11. Eu gostei dos gráficos da aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

12. Eu gostaria de utilizar com mais frequência esse tipo de aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

13. Eu me ajustei facilmente ao dispositivo de controle usado para interagir com o ambiente.*

1 2 3 4 5

Discordo Concordo Fortemente

14. Eu achei muito adequado o dispositivo de interação escolhido para essa aplicação.*

1 2 3 4 5

Discordo Concordo Fortemente

15. Eu me senti confortável com a simulação.*

1 2 3 4 5

Discordo Concordo Fortemente

16. Eu me senti envolvido com a simulação.*

1 2 3 4 5

Discordo Concordo Fortemente

17. A representação visual parece bastante com a realidade.*

1 2 3 4 5

Discordo Concordo Fortemente

18. Eu me senti tranquilo e à vontade ao utilizar o sistema.*

1 2 3 4 5

Discordo Concordo Fortemente

19. A forma de navegação é fácil .*

1 2 3 4 5

Discordo Concordo Fortemente

20. Acho que ainda vou lembrar-me de como usar o sistema, mesmo daqui a uma semana.*

1 2 3 4 5

Discordo Concordo Fortemente

21. Eu achei que o uso do dispositivo na aplicação possibilita melhora da interação.*

1 2 3 4 5

Discordo Concordo Fortemente

22. Você achou que o uso desse tipo de dispositivo tem a capacidade de melhorar o uso da aplicação?*

1 2 3 4 5

Discordo Concordo Fortemente

23. Alguma Sugestão?

ANEXO D - Termo de Consentimento Assinado pelos Participantes dos Testes.**TERMO DE CONSENTIMENTO**

Declaro, por meio deste termo, que concordei em participar na pesquisa de campo referente ao projeto/pesquisa intitulado(a) “ **Uso de Interação Natural em Simulador de Procedimentos para Treinamento no Setor Elétrico**” desenvolvida(o) por **Rafael Henrique Assis de Castro**. Fui informado(a), ainda, de que a pesquisa é coordenada / orientada por **Iguatemi Eduardo da Fonseca** e **Tatiana Aires Tavares**.

Afirmo que aceitei participar por minha própria vontade, sem receber qualquer incentivo financeiro ou ter qualquer ônus e com a finalidade exclusiva de colaborar para o sucesso da pesquisa. Fui informado(a) dos objetivos estritamente acadêmicos do estudo, que, em linhas gerais é: “o desenvolvimento de um simulador de treinamento de procedimento no setor elétrico utilizando dispositivo de reconhecimento de gestos”.

Minha colaboração se fará de forma anônima, por meio de observação e coleta de dados através de formulários e análise da(s) utilização(ões) do simulador. Fui ainda informado(a) de que posso me retirar desse(a) estudo a qualquer momento, sem sofrer quaisquer sanções ou constrangimentos.

Atesto recebimento de uma cópia assinada deste Termo de Consentimento Livre e Esclarecido.

Assinatura do(a) testemunha(a)

Assinatura do pesquisador

Assinatura do(a) participante

Campina Grande, ____ de _____ de ____

ANEXO E - Respostas do Questionário Pré-teste.

Questão:	1					2					3					4					5					6					7				
	usuário	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e	a	b	c	d	e	a	b	c	d
1		X					X				X				X	X	X			X						X							X		
2				X				X			X				X					X			X				X					X			
3		X					X				X						X			X						X						X			
4		X						X			X					X	X	X			X					X					X				
5		X					X					X				X	X																X		
6		X					X				X	X	X	X	X	X	X	X	X	X	X				X					X					
7		X					X				X		X	X	X	X	X	X												X					
8		X					X				X	X				X						X				X				X					
9	X						X				X	X	X	X	X	X	X	X							X					X				X	
10		X					X				X	X	X	X	X	X	X				X								X						
11		X					X				X		X	X	X	X									X								X		
12		X					X				X		X	X	X	X														X					
13		X				X					X	X		X	X	X				X	X			X	X					X				X	
14		X				X						X	X	X	X	X	X	X	X	X	X			X	X			X						X	
15		X					X				X			X	X	X	X	X						X			X				X				
16	X					X					X		X	X	X	X	X	X	X	X	X			X			X			X					
17		X					X				X		X	X	X	X	X	X	X	X	X			X			X							X	
18		X					X				X			X	X	X														X					
19		X					X				X		X	X	X	X	X	X	X	X	X			X	X		X			X					
20		X					X				X		X	X	X	X	X	X	X	X	X			X	X			X			X				
21			X					X			X				X									X		X				X					
22		X					X				X		X	X	X			X	X	X	X			X											X
23		X					X					X	X	X	X	X	X	X	X	X	X			X						X					X
24		X					X				X			X	X	X	X	X	X	X	X			X			X				X				

ANEXO F - Respostas do Questionário do Uso do ATreVEE 3D.

Usuário	Questões																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	5	5	5	4	5	5	5	4	4	5	5	4	5	4	5	5	5	4	5	5	4	5
2	4	4	5	5	5	5	4	5	5	5	5	5	4	5	4	5	5	5	5	5	5	5
3	2	5	3	4	4	4	4	4	5	4	5	4	2	4	4	4	4	4	4	4	5	4
4	4	4	4	4	4	4	3	4	5	4	4	4	4	4	4	4	4	4	4	3	4	4
5	4	4	3	2	3	4	2	4	4	3	4	3	4	4	4	2	3	4	4	4	4	4
6	5	5	5	3	3	4	5	4	3	2	2	3	5	3	3	2	3	3	4	4	4	4
7	5	4	5	5	5	4	4	5	5	4	4	5	5	5	5	4	5	5	5	5	5	5
8	4	3	4	3	4	4	3	4	5	5	3	5	4	4	4	4	4	4	5	5	5	5
9	5	5	5	4	4	4	3	4	4	4	5	5	5	4	4	4	4	5	5	5	3	4
10	5	4	3	3	4	5	4	5	5	5	4	5	3	4	4	5	4	3	4	5	4	4
11	4	4	3	4	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
12	5	5	5	4	5	5	5	5	5	5	5	4	5	5	5	5	5	4	5	5	5	5
13	4	5	4	4	5	5	4	5	5	5	5	5	4	5	4	5	5	5	4	5	5	5
14	3	3	3	4	2	5	5	5	5	3	4	4	4	2	4	2	4	5	4	4	3	4
15	5	5	4	5	4	4	5	5	5	4	3	5	5	5	5	4	3	5	5	5	5	4
16	4	5	5	4	3	4	4	5	5	4	4	5	5	3	5	4	3	5	5	5	5	5
17	4	4	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5
18	4	3	5	2	5	5	3	5	5	5	5	5	4	4	4	4	5	4	4	5	4	5
19	4	5	5	4	4	5	3	5	5	4	4	5	5	4	5	4	3	5	5	5	4	4
20	5	5	5	4	5	4	4	5	4	4	4	4	4	4	4	4	4	4	5	5	5	5
21	5	5	5	5	3	3	5	4	3	3	3	4	4	4	4	4	3	4	4	5	4	4
22	4	5	5	3	3	4	2	5	5	5	4	5	3	5	5	5	5	5	3	5	5	5
23	5	5	4	4	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5
24	5	5	5	5	4	5	4	5	5	5	5	4	5	4	5	4	4	5	5	5	4	5
TOTAL MOUSE	4.33	4.45	4.37	3.91	4.12	4.45	3.95	4.66	4.66	4.29	4.16	4.5	4.33	4.2	4.41	4.12	4.16	4.45	4.54	4.75	4.45	4.58

ANEXO G - Respostas do Questionário do Uso do ATreVEE IN.

Usuário	Questões																					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	4	4	4	4	4	3	5	4	4	4	3	4	4	4	2	4	5	4	4	5	4	4
2	3	3	2	4	4	5	5	5	5	5	4	5	2	4	3	4	5	5	3	5	4	5
3	5	4	4	3	4	3	4	4	4	3	4	3	3	3	3	3	3	3	3	3	4	4
4	3	3	3	3	4	4	4	4	4	3	4	4	3	3	3	4	4	3	4	3	4	4
5	1	1	1	1	3	5	4	4	4	2	4	2	2	2	2	1	3	2	2	2	5	4
6	4	4	2	2	4	4	4	4	4	5	4	4	3	5	4	4	4	3	3	5	4	5
7	4	5	4	4	4	4	5	5	5	5	5	5	4	5	5	5	5	5	4	5	5	5
8	1	1	1	2	1	3	3	2	2	2	2	1	2	2	2	2	3	2	2	4	3	3
9	2	3	5	2	4	1	4	4	5	5	4	4	2	4	2	4	4	3	2	5	3	4
10	4	4	2	2	3	4	5	5	4	5	4	4	2	3	4	5	4	3	3	4	3	4
11	3	4	4	3	4	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	5	5
12	3	5	4	3	5	3	5	5	5	5	5	4	4	5	4	5	5	5	4	5	5	5
13	4	5	5	4	4	4	4	4	5	5	5	5	4	5	5	5	5	5	4	5	5	5
14	2	4	2	2	2	2	5	5	5	3	4	4	2	3	2	2	4	5	3	5	2	5
15	5	5	4	3	4	4	5	5	4	5	3	5	5	4	5	5	4	4	5	5	5	5
16	3	3	5	3	4	4	5	5	4	5	5	5	4	3	4	5	5	4	2	5	4	5
17	5	4	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	4	5	5	5
18	4	4	4	4	4	4	5	5	5	5	5	5	5	5	4	5	5	5	4	4	5	5
19	5	5	5	4	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
20	2	4	2	3	4	5	5	5	4	4	5	5	2	4	3	4	4	2	5	5	5	5
21	4	4	3	4	4	3	4	5	5	5	4	4	3	3	3	5	3	4	4	5	5	5
22	3	4	3	3	4	4	5	5	5	5	4	4	3	4	4	4	5	5	3	5	5	5
23	3	5	5	5	5	5	5	5	5	5	4	5	5	5	5	5	5	5	4	5	5	5
24	5	5	5	5	3	4	5	5	5	5	5	5	5	5	5	5	5	5	4	5	5	5
TOTAL LEAP	3.41	3.87	3.5	3.25	3.83	3.87	4.62	4.58	4.5	4.41	4.2	4.25	3.45	4	3.7	4.2	4.37	4.04	3.58	4.58	4.37	4.66