

Avaliação de uma Luva de Dados em um Sistema Virtual para Aplicações de Treinamento Médico

Adriano Bezerra¹, Fátima L. S. Nunes², Cleber Gimenez Corrêa³

¹Instituto de Ciências Matemáticas e de Computação – ICMC – Universidade de São Paulo (USP) São Carlos – SP – Brasil

²Escola de Artes, Ciências e Humanidades – EACH – Universidade de São Paulo (USP) São Paulo – SP – Brasil

³Escola Politécnica – Universidade de São Paulo (USP) São Paulo – SP – Brasil
adriano@icmc.usp.br, fatima.nunes@usp.br, correacleber@yahoo.com.br

Abstract. *Virtual Reality applications that use datagloves can offer more suitable tools for professionals and students from health area, once from the five senses of humans, only the touch is bi-directional. The touch allows acquire and input information in Virtual Environments, like an action that requires a deformation of an object, for instance. In this paper the insertion of a dataglove and an evaluation of its use in a Virtual Reality Framework that provides facilities to implement medical training tools are presented.*

Resumo. *Aplicações de Realidade Virtual (RV) que utilizam luva de dados podem oferecer ferramentas mais adequadas para profissionais e estudantes da área de saúde, uma vez que, dos cinco sentidos que os humanos possuem, somente o tato é bidirecional. Com este sentido, pode-se tanto adquirir informações de um Ambiente Virtual quanto provocar deformações no mesmo por meio de aplicação de força, fornecendo ao usuário uma melhor sensação de imersão no ambiente. Neste artigo é apresentada a inclusão de uma luva de dados e uma avaliação do seu uso em um Framework de Realidade Virtual destinado a auxiliar no processo de implementação de ferramentas para treinamento médico.*

1. Introdução

Nos últimos tempos, tem-se observado diversos trabalhos com o propósito de construir sistemas computacionais para treinamento de estudantes e profissionais da área médica, visando ao aprendizado mais efetivo de procedimentos com menor custo. Neste contexto, existem Ambientes Virtuais (AVs) construídos com técnicas de Realidade Virtual (RV), que podem suportar dispositivos convencionais como *mouse*, teclado e monitores, e dispositivos não convencionais, como luva de dados e equipamento háptico, permitindo a interação entre ser humano e máquina com uma elevada imersão.

Os ambientes criados com técnicas de RV podem oferecer interação mais avançada entre homem e máquina, sendo de grande auxílio no treinamento médico [Oliveira *et al.*, 2007]. Uma das vantagens de aplicações de RV para o treinamento médico é a possibilidade de permitir que o estudante execute virtualmente um

determinado procedimento antes de executá-lo em um paciente real. Além disso, a RV proporciona ao cirurgião a prática do aprendizado e de detalhes da cirurgia em um Ambiente Virtual (AV), podendo fornecer um *feedback* de sua atuação conforme a prática adquirida no aprendizado [Liu *et al.*, 2003].

As aplicações de RV para treinamento médico são também ferramentas educativas, pois, além de praticar o procedimento, o usuário pode visualizar objetos modelados tridimensionalmente que representam órgãos humanos, parcial ou totalmente, podendo acessar estruturas internas e mudar o ponto de vista desejado [Souza *et al.*, 2002]. Assim, tais aplicações podem oferecer uma nova forma de aprendizado, visto que imagens tridimensionais, exploração interativa e informações táteis podem ser combinadas para oferecer uma simulação mais realista [Machado, 2003].

A exatidão é essencial e a prática proporciona benefícios significativos, tanto na recuperação do paciente quanto na obtenção de resultados esperados. Assim, características como a modelagem adequada de objetos tridimensionais, a reprodução de ações executadas e as reações possíveis (deformação do órgão, perfuração de diversos tipos de tecidos) do exame a ser simulado precisam ser levadas em consideração durante a elaboração do AV [Corrêa *et al.*, 2007].

Este artigo apresenta a inserção de uma luva de dados em um *Framework* de RV, que oferece suporte a implementações de ferramentas para treinamento médico, além da avaliação da aplicação realizada por profissionais da área médica. Para atingir este objetivo, a seção 2 apresenta alguns conceitos sobre interação para melhor entendimento do contexto. A seção 3 descreve o *Framework* utilizado para o desenvolvimento da interação. Na seção 4, é apresentada a metodologia utilizada para inserir a luva de dados no *Framework*, empregando a integração entre linguagens de programação. A seção 5 mostra os resultados e as discussões da avaliação realizada com profissionais da área médica, levando em consideração aspectos computacionais e humanos. E, ao final, a seção 6 apresenta as conclusões.

2. Conceitos de Interação

Em um sistema de RV, a interação é responsável pela transferência ou mapeamento das entradas em relação à localização e orientação do usuário. Por isso, a manipulação dos objetos deve ocorrer de forma natural e o controle dos movimentos do usuário deve ser intuitivo e contínuo, por exemplo, com a utilização da força e do tato para o apoio a algumas tarefas [Bowman *et al.*, 2004].

Uma técnica de interação inclui tanto componentes de *hardware* (dispositivos de entrada e saída) quanto de *software*. As técnicas podem ser classificadas em três categorias: **navegação**, que trata do movimento do usuário dentro do AV durante uma simulação; **seleção** e **manipulação**, que consiste, respectivamente, na escolha de algum objeto virtual e na modificação de suas características; e **orientação**, que se relaciona ao controle do sistema, envolvendo comandos específicos disponíveis para alterar o estado do mesmo [Bowman *et al.*, 2002].

Para explorar e manipular os objetos no mundo real, o usuário pode usar suas mãos, sendo esta uma forma natural de interação com o AV [Hsu, 2008]. Deste modo,

luvas de dados são desejáveis nessas aplicações, em virtude de permitirem que os movimentos das mãos do usuário sejam capturados e refletidos no AV. É desejável também que o usuário possa visualizar uma representação de sua mão no ambiente, por meio de modelos tridimensionais, a fim de que possa se sentir imerso e se localizar no AV.

Neste trabalho, foram consideradas simulações de exames de biópsia para o treinamento médico usando RV. O procedimento consiste em usar a luva de dados para simular o ato de segurar um objeto, o qual representa um órgão humano, e então, manipular uma seringa virtual, até atingir um determinado alvo (nódulo ou outra estrutura de interesse, por exemplo). Em um AV imersivo para simulação, cada uma dessas fases merece especial atenção em relação à interação do usuário com a aplicação. As aplicações foram construídas usando um *Framework* de RV, apresentado a seguir.

3. O *Framework ViMeT*

O *Virtual Medical Training (ViMeT)* é um *Framework* de RV orientado a objetos que utiliza as tecnologias Java e C++, juntamente com as *Application Programming Interfaces (APIs)* Java 3D e *Java Native Interface (JNI)* [Oliveira *et al.*, 2007]. Esse *Framework* está sendo desenvolvido para auxiliar a geração de aplicações de treinamento médico, disponibilizando funcionalidades classificadas como importantes para tais simulações: interface gráfica, detecção de colisão, deformação, interação com equipamentos convencionais e não convencionais, estereoscopia, importação e modelagem de objetos 3D, além da geração de AVs. A Figura 1 apresenta um AV gerado por meio do ViMeT.

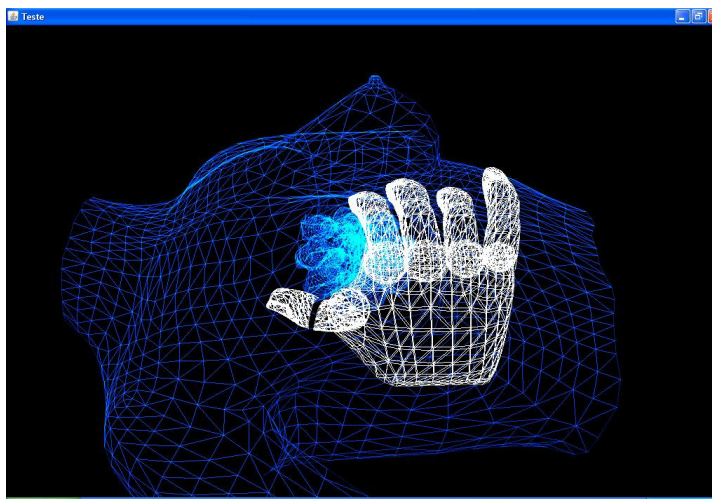


Figura 1. Exemplo de um Ambiente Virtual gerado pelo ViMeT

Para a importação dos objetos virtuais, o usuário pode selecionar os objetos desejados durante a geração do AV, definindo características como posição, rotação e tamanho desses. A posição da mão virtual é definida de forma fixa na geração da aplicação pelo ViMeT sem a intervenção do usuário nesta etapa. É possível, no entanto, alterar tais parâmetros na aplicação gerada.

Apesar de o presente trabalho apresentar somente a interação com a luva de dados, o ViMeT também possibilita a utilização de outros dispositivos, tais como:

mouse, teclado e equipamento háptico [Corrêa *et al.*, 2008b]. A Figura 2 mostra o diagrama de classes do *Framework* indicando em cor mais escura as classes implementadas no presente trabalho.

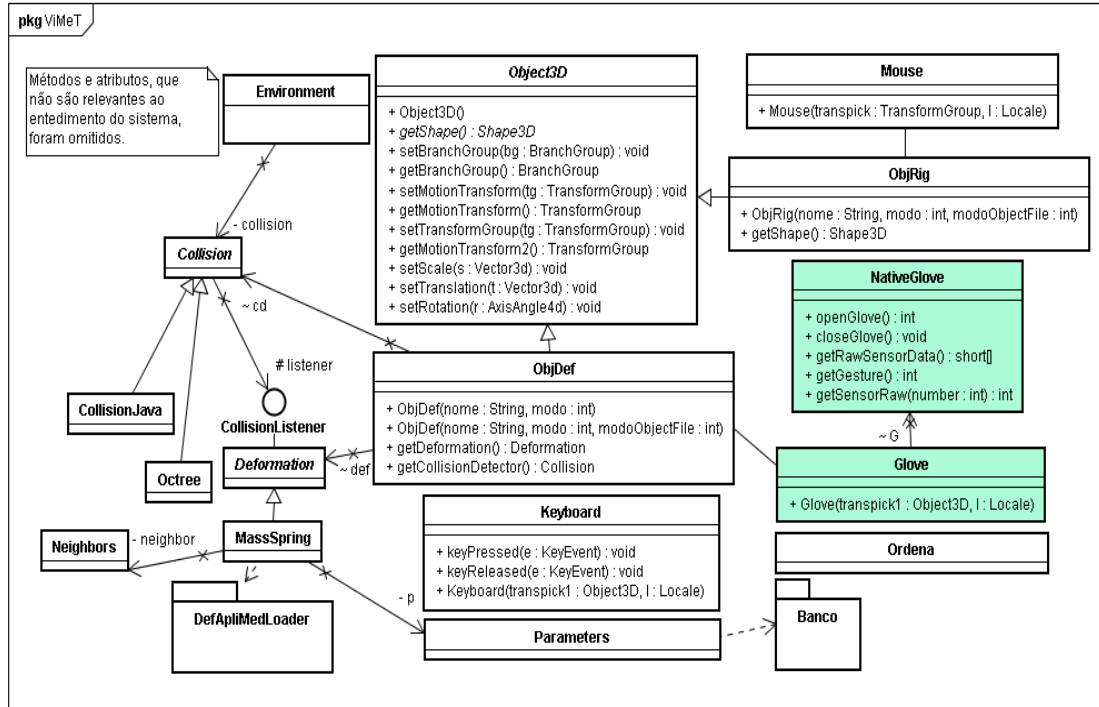


Figura 2. Diagrama de classes do ViMeT.

4. Inclusão da Luva de Dados no *Framework*

A luva de dados utilizada no ViMeT é uma 5DT *DataGlove Ultra*, que contém 5 sensores de fibra ótica, localizados na região dos dedos, mostrada na Figura 3. Esses sensores permitem a captação da informação sobre a flexão das falanges dos dedos, possibilitando a captura de movimentos, que são transmitidos à aplicação [5DT – *Fifth Dimension Technologies* 2009].

As informações provenientes da luva de dados em relação à flexão das falanges são transmitidas ao sistema, fazendo com que os dedos representados na mão virtual se movam corretamente e no ângulo desejado. A flexão dos dedos indica o ato de segurar e soltar o órgão humano pelo usuário durante a simulação.

Neste projeto foi utilizada uma interface entre a aplicação em Java e as bibliotecas e *drivers* em C/C++, visto que a aplicação gerada pelo ViMeT é escrita em linguagem Java e os *drivers* e as bibliotecas com funções para operações com a luva de dados são escritos em linguagens de programação C/C++.

A interface estabelecida por meio da integração das linguagens elimina alguns problemas, mas requer o conhecimento das linguagens de programação adotadas, bem como uma forma de integração entre elas. Um possível problema pode ser o tempo de resposta devido à nova camada de *software* para realizar a integração, o que pode causar atrasos em sistemas interativos [Corrêa *et al.*, 2008b]. No entanto, essa possibilidade de limitação não foi confirmada na avaliação do sistema, como será apresentado adiante.

Foi utilizada a API JNI, que integra o JDK (*Java Development Kit*) e possibilita a interoperabilidade entre programas escritos em Java com outras linguagens, como C/C++ [Sun, 2009], conforme detalhado no próximo tópico.

4.1. Integração do equipamento - aspectos técnicos

As etapas de implementação do código em linguagem de programação C++ para a luva de dados são realizadas de acordo com o diagrama da Figura 4, com as funções fornecidas pela biblioteca do dispositivo, as quais são acionadas por métodos implementados na aplicação usando linguagem Java. Primeiramente é feita a verificação das portas USB, utilizando-se o comando *fdScanUSB*, que realiza uma busca a fim de verificar em qual porta a luva está conectada. Em seguida, ocorre a definição do tipo de luva de dados conectada, levando em consideração o número de sensores e se o dispositivo em questão é para a mão esquerda ou para a mão direita.



Figura 3. 5DT Data Glove 5 Ultra [Fifth Dimension Technologies, 2009].

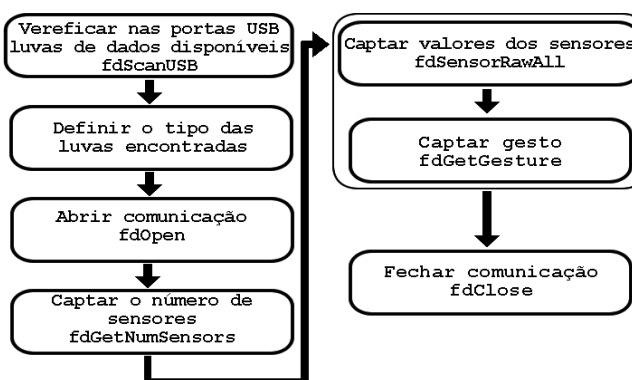


Figura 4. Diagrama de execução da luva de dados.

Depois de definido o tipo de luva, a função *fdOpen* permite o estabelecimento da comunicação da aplicação com o dispositivo, armazenando o endereço que representa a luva de dados no ponteiro *pGlove*. Caso ocorra um erro na inicialização do equipamento, uma mensagem é apresentada ao usuário.

A função *fdGetNumSensors* é responsável pela obtenção do número de sensores do dispositivo, uma vez que o *software* pode ser utilizado por outras luvas de dados do mesmo fabricante. Para captar os valores de todos os sensores do equipamento simultaneamente, é empregada a função *fdGetSensorRawAll*.

Por fim, a função *fdGetGesture* é empregada para obter o número de um gesto definido pela posição dos dedos e a função *fdClose* é usada para fechar a comunicação com a luva de dados, recebendo como parâmetro o ponteiro *pGlove*.

4.2. Implementação da mão virtual - aspectos técnicos

Em relação à representação da mão do usuário no AV, foi cuidadosamente planejada a modelagem do objeto tridimensional, uma vez que a luva de dados utilizada não possui mais sensores para fazer a representação correta de cada falange dos dedos.

A modelagem da mão virtual foi projetada seguindo uma estrutura hierárquica, na qual os membros da mão foram divididos e interligados, visando à obtenção de uma maior sensação de realismo [Corrêa *et al.*, 2008]. Os membros foram divididos em 3

partes: palma da mão, corpos dos dedos e pontas dos dedos. O corpo de cada dedo foi acoplado à palma da mão e a ele foi ligada a ponta de cada dedo. Se o usuário manipular a palma da mão, todas as outras partes dos dedos se movimentarão. O mesmo acontece quando ocorrer a movimentação do corpo de um dedo, sendo que a ponta do mesmo acompanhará o movimento.

Após a execução de testes, verificou-se que os dedos modelados não rotacionavam corretamente. Para resolver a questão, foram inseridas esferas virtuais como guia de rotação para cada um dos dedos virtuais, permitindo que o movimento se limitasse ao centro de cada uma delas, propiciando um maior realismo, conforme representado na Figura 5.

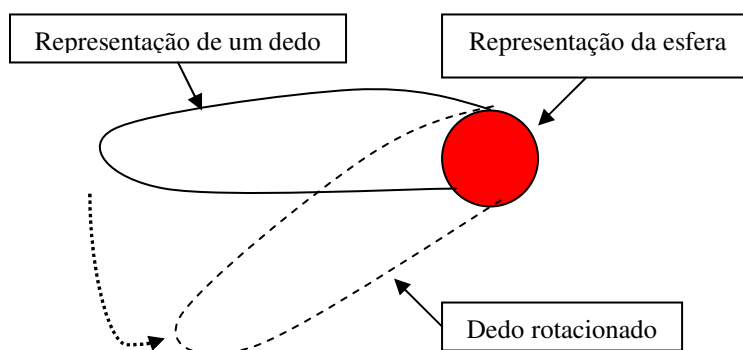


Figura 5. Representação do movimento dos dedos limitados à esfera.

Com essa modelagem utilizando uma estrutura hierárquica e considerando que a luva de dados possui somente um sensor por dedo, decidiu-se fazer as modelagens dos dedos de forma a representar um leve grau de flexão das falanges mais distantes da esfera ao invés de os modelar em linha reta. Tal procedimento também foi realizado para propiciar um maior realismo. A flexão é mostrada na Figura 6, apresentando a modelagem de um único dedo.

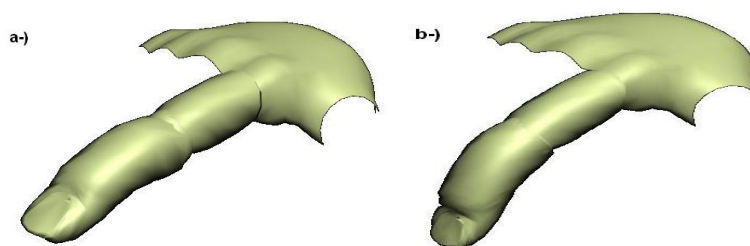


Figura 6. (a) Modelagem de um dedo em linha reta. (b) Modelagem de um dedo com leve flexão das falanges

A API Java 3D constrói AVs usando o conceito de grafo de cena. O grafo de cena é uma forma de representação de objetos nas cenas, provendo uma camada de abstração hierárquica desses objetos. Os nós folhas, em geral, contêm a descrição geométrica de um objeto. Nós internos comumente representam transformações tridimensionais como rotação, translação e escala. O nó raiz, que se conecta aos demais, direta ou indiretamente, é a representação do AV como um todo [Silva *et al.*, 2004].

A partir das implementações realizadas, algumas alterações foram necessárias no grafo de cena do ViMeT, a fim de considerar as esferas e os dedos virtuais. Na Figura 7,

é apresentado o grafo de cena atual do *Framework*, indicando o posicionamento dos objetos que representam a mão virtual. No *BranchGroup* BG2 (identificado como o ponto 1 na Figura), foram acrescentados outros objetos do tipo *BranchGroups*, : o BGz (ponto 4 na Figura) refere-se ao objeto que representa a ponta do dedo, o BGy (ponto 3 na Figura) refere-se ao corpo do dedo e o BGx (ponto 2) indica a esfera de rotação. Este último está ligado ao nó BG2, aludindo à palma da mão.

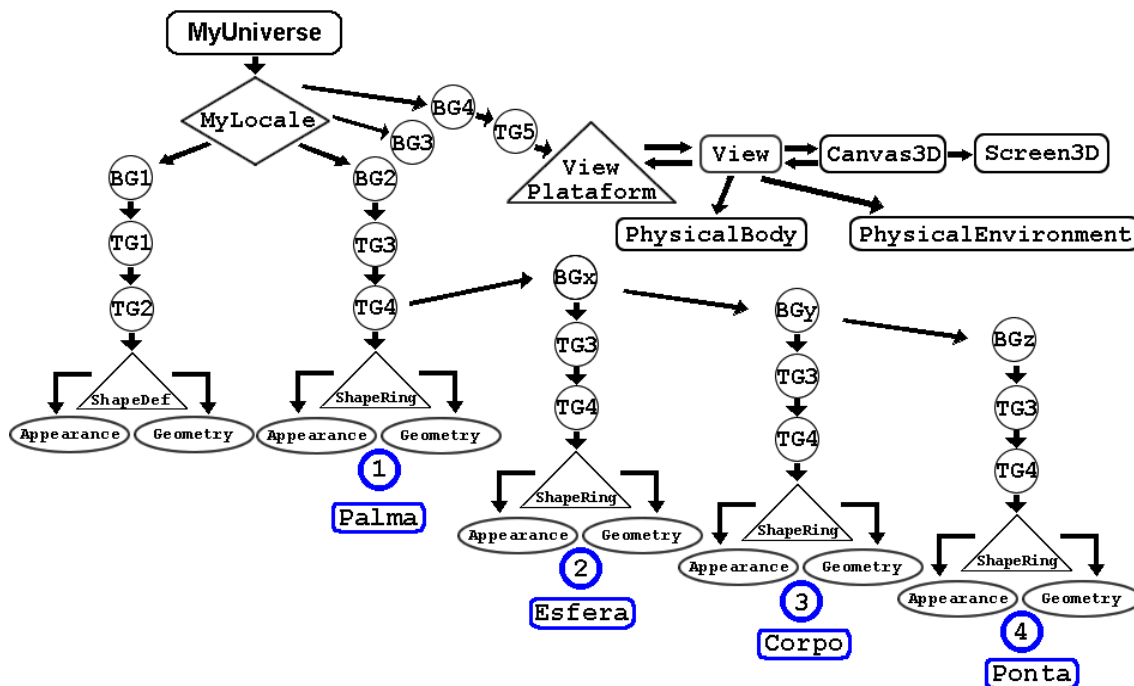


Figura 7. Divisão da mão modelada no grafo de cena.

4.3. Avaliação do Sistema

A avaliação da interação em aplicações construídas com o *Framework* ViMeT foi uma fase de grande importância para o aperfeiçoamento do sistema. Esta atividade procurou levar em consideração aspectos técnicos, mensurando o tempo de resposta diante de um comando do usuário e do número de quadros exibidos por segundo.

Além disso, o tipo de dispositivo utilizado é determinante porque envolve questões como: quantidade de graus de liberdade, influência na eficiência do processamento e o grau de realismo exercido. Em relação ao processamento, neste trabalho, pode-se destacar também a camada adicional para a integração entre as linguagens de programação Java e C++.

Outra questão relacionada à avaliação diz respeito aos aspectos humanos, tais como conforto com os dispositivos, satisfação do usuário, tempo de execução de uma determinada tarefa, experiências anteriores com aplicações de RV e com o procedimento médico a ser executado (exames de punção).

A avaliação do sistema foi executada em parceria com uma Faculdade de Medicina, tendo participado um professor e onze alunos de Medicina. Foi realizada uma descrição do processo de avaliação para os voluntários definindo-se suas etapas e

objetivos, com explicações sobre conceitos de RV, bem como algumas peculiaridades do sistema e dos dispositivos. Os resultados são apresentados na próxima seção.

5. Resultados e Discussões

A inclusão de uma luva de dados no *Framework ViMeT* por meio do recurso JNI colaborou no avanço da literatura em se tratando de RV, visto que não foi encontrado material disponível com detalhes de implementação na literatura da área. No início, havia certa preocupação quanto ao desempenho da aplicação, devido à integração entre as linguagens Java, C++ e o processamento das informações dos equipamentos não convencionais. Entretanto, os testes mostraram um nível satisfatório de interação em relação ao tempo de resposta.

Durante a execução do procedimento, o tempo que a aplicação levou para apresentar um quadro variou de 16,39 a 17,54 milissegundos e o número correspondente de quadros por segundo variou de 57 a 61. Segundo Kirner (2006), em RV, uma taxa de quadros por segundo ideal deve ser igual ou superior a 20 para manter a ilusão de movimento, sendo que uma taxa de 8 a 10 quadros por segundo é considerada aceitável para propiciar a imersão.

A avaliação foi realizada com as doze pessoas, sendo um professor e onze alunos de Medicina, e comprovou a preferência dos usuários pela luva de dados em comparação ao teclado, bem como a sua usabilidade, conforme pode ser observado no gráfico apresentado na Figura 8. A maioria dos voluntários afirmou que a luva de dados é mais intuitiva do que o teclado. Neste contexto, a luva de dados traz uma maior usabilidade em comparação ao teclado, entretanto, o teclado tem um custo muito menor se comparado à luva de dados.

Nesta avaliação foi aplicado um questionário antes dos voluntários utilizarem o sistema, que abordava questões sobre o conhecimento em relação à área de RV. Todos os entrevistados conheciam a área, dominando, inclusive, conceitos sobre algumas definições e aplicações, sendo que dois deles já tinham usado equipamentos de RV. No entanto, nenhum voluntário tinha até aquele momento trabalhado ou participado de simulações que envolviam luvas de dados.

Os estudantes que participaram das simulações de treinamento tiveram dificuldades para perceber a posição dos objetos no AV, principalmente em consideração à percepção de profundidade. Uma proposta sugerida por eles foi a inclusão de outros pontos de vista na tela da aplicação, a fim de que o usuário pudesse ter maior noção de espaço dentro do AV.

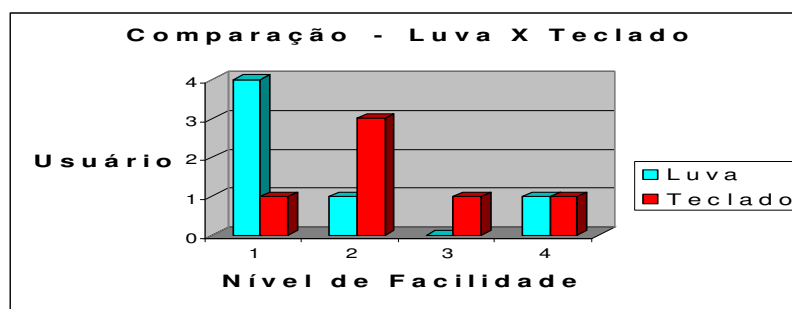


Figura 8 – Comparação da usabilidade dos dispositivos.

6. Conclusões

Neste trabalho foi apresentado o processo completo da inclusão, implementação e avaliação de uma luva de dados em um módulo de interação de um *Framework*, que tem o objetivo de gerar aplicações para treinamento médico virtual.

O primeiro ponto a ser destacado como contribuição é que a integração entre linguagens de programação ainda é pouco detalhada na literatura existente; com a implementação apresentada, é possível auxiliar no desenvolvimento de aplicações desta natureza, inclusive para outros Sistemas Operacionais, aproveitando a portabilidade da linguagem de programação Java e reduzindo os custos de desenvolvimento.

Como a simulação de exames de biópsia envolve os atos de segurar o órgão humano e manipular um instrumento médico para coletar material celular, se faz necessário, além de equipamentos como luva de dados e teclado, dispositivos como *mouse* e equipamento háptico, que propiciam o controle de instrumentos de coleta, permitindo no caso deste último dispositivo um *feedback* tátil quando há a ocorrência de colisão entre objetos virtuais (instrumento médico e órgão humano), indicando a perfuração da pele e demais tecidos do órgão [Corrêa *et al.*, 2009].

Para a área médica, o *Framework* empregado propicia a geração de aplicações para treinamento de exames de biópsia, utilizando dispositivos convencionais e não convencionais, oferecendo opções para inclusão de dispositivos de acordo com a disponibilidade financeira e o grau de realismo desejado. A estrutura atual permite que aplicações para simular outros procedimentos médicos possam ser geradas com as devidas adaptações.

Os sistemas de RV apresentam diversos outros benefícios, além da diminuição ou eliminação de riscos no treinamento, como a disponibilidade contínua do AV, a capacidade de simular diversos casos (diferentes anomalias, por exemplo), a minimização do uso de cobaias e a possibilidade do profissional da área verificar seu desempenho em todas as etapas de um determinado procedimento médico. Desta forma, a aquisição de habilidade para executar certos procedimentos pode ser facilitada, assim como a aquisição de experiência, tornando o usuário especialista na realização dos procedimentos e aumentando a confiabilidade do processo.

Referências

- Bowman, D.; Kruijff, E.; Laviola, J.; Poupyrev, I. In: *3D User Interfaces: Theory and Practice*, Addison Wesley, July 2004.
- Bowman, D.; Gabbard, J.; Hix, D. (2002). *A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods*. In: *Teleoperators and Virtual Environments*, vol. 11, no. 4, p. 404-424.
- Corrêa, C. G.; Nunes, F. L. S.; Bezerra, A. Implementação de Interação sem Sistemas Virtuais para Simulação de Exame de Biopsia In: Anais do XXVIII Congresso da SBC – WIM – Workshop de Informática Médica, Belém do Pará, PA, Brasil, 2008. p. 91-100.
- Corrêa, C. G.; Bezerra, A.; Nunes, F. L. S. Integração de Linguagens de Programação para Uso de Dispositivos Não-Convencionais: Possível Solução para Construir

- Aplicações com Baixo Custo. In: *X Symposium on Virtual and Augmented Reality*, João Pessoa, PB, Brasil, 2008. *Proceedings of the X Symposium on Virtual and Augmented Reality*, João Pessoa, PB, Brasil, 2008. p. 266-269(b).
- Corrêa, C. G.; Nunes, F. L. S.; Bezerra, A.; Carvalho Junior, P. M. *Evaluation of VR Medical Training Applications under the Focus of Professionals of the Health Area*. In: *The 24th Annual ACM Symposium on Applied Computing*, Honolulu, Hawaii, USA, 2009. *Proceedings of the 24th Annual ACM Symposium on Applied Computing*. New York, USA: Association of Computer Machinery, 2009. v. 1. p. 821-825.
- Hsu, J. (2009) *Active Interaction Devices*. <http://www.hitl.washington.edu/scivw/EVE/I.D.1.a.ActiveInteraction.html>, Janeiro.
- Kirner, C. (2009). *Sistemas de Realidade Virtual*. Universidade Federal de São Carlos – UFSCar, São Carlos, SP, Brasil. <http://www.dc.ufscar.br/~grv/tutrv/tutrv.htm>, Fevereiro.
- Liu, A.; Tendick, F.; Cleary, K.; Kaufmann, C. (2003) “*A Survey of Surgical Simulation: Application, Technology and Education*”. In: MIT Press, vol. 12, Dezembro.
- Machado, L. S.; Zuffo, M. K. “*Development and Evaluation of a Simulator of Invasive Procedures in Pediatric Bone Marrow Transplant*. *Studies In Health Technology and Informatics*. Amsterdam, v.94, p. 193 – 195, 2003.
- Oliveira, A. C. M. T. G. de.; Nunes, F. L. S.; Bezerra, A. *ViMet – Concepção e Implementação de um Framework para simulação de exames de punção usando Realidade Virtual*. In: *Anais do Symposium on Virtual Reality*, Petrópolis, RJ, 2007.p. 253-262.
- Sun (2009) *Java Native Interface Specification*, <http://java.sun.com/j2se/1.4.2/docs/guide/jni/spec/jniTOC.html>, Março.
- Silva, R. J. M. da.; Raposo, A. B.; Gattas, M. “*Grafo de cena e realidade virtual*”. No. 11/04. In: *Monografias em Ciência da Computação*. Rio de Janeiro - RJ: PUC-Rio, 2004.
- Souza, L. A.; Junior, C. S.; Zuffo, M. K. (2002) “*Desenvolvimento e avaliação de um simulador de realidade virtual para treinamento de biópsia da tireóide*”. In: *Anais do XVIII Congresso Brasileiro de Engenharia Biomédica*, São José dos Campos – SP.
- 5DT - *Fifth Dimension Technologies* (2009) <http://www.5dt.com/hardware.html>, Fevereiro.