

Integração de uma Luva de Dados a um Framework de Realidade Virtual

Rubens G. S. Correia
LabTEVE
Universidade Federal da Paraíba (UFPB)
João Pessoa/PB - Brasil
e-mail: rubens.jp.br@gmail.com

Liliane S. Machado
LabTEVE / Departamento de Informática
Universidade Federal da Paraíba (UFPB)
João Pessoa/PB - Brasil
e-mail: liliane@di.ufpb.br

Abstract— This paper discusses the incorporation of a new functionality in a Virtual Reality framework, the integration of a data glove, showing the project stages and the tests that were done. It will be shown how the classes that support the data glove are attached to the framework, inheriting from generic classes that deal with the devices, the support to composite models, a necessity that appeared along the project and the results obtained through these new features.

Keywords— Realidade virtual, luva de dados, modelos compostos, desenvolvimento de aplicações.

I. INTRODUÇÃO

A Realidade Virtual (RV) tem sido bastante utilizada em simulações e treinamentos nas mais diversas áreas. Por meio do uso destas aplicações em ambientes virtuais o usuário é levado a uma nova realidade a qual é submetido a experiências similares as que ele virá a ter em sua vida profissional. Para interagir nestes ambientes tridimensionais, dispositivos de interação espacial, como rastreadores e luvas de dados, são necessários. A interação proporcionada por estes dispositivos garantem também uma maior imersão levando o usuário a uma experiência ainda mais realista. O sentido da visão costuma ser preponderante em aplicações de realidade virtual, mas os outros sentidos, como tato, audição, etc. também podem ser usados para enriquecer a experiência do usuário [1].

Para acelerar o processo de desenvolvimento de aplicações de RV existem *frameworks* que oferecem um conjunto de recursos já prontos, aumentando assim a produtividade de seus usuários. Deste modo, a integração de novas funcionalidades a estes pacotes permite a expansão de sua potencialidade de uso.

O presente trabalho trata do processo de integração de uma luva de dados em um *framework* de RV, acrescentado-lhe assim uma nova forma de interação e imersão. Neste processo observou-se a necessidade de uso de objetos hierárquicos.

II. LUVAS DE DADOS E AMBIENTES DE RV

A luva de dados é um dispositivo não convencional de interação semelhante a uma luva comum que possibilita adquirir informações a partir dos movimentos dos dedos da

mão. Para determinar esses movimentos são utilizados, em geral, sensores mecânicos ou fibra ótica. Os sensores de fibra ótica são utilizados nas versões mais populares de luvas de dados e consistem de um fio de fibra ótica com junções. Quando uma junta é movida o cabo dobra-se, reduzindo a passagem de luz por ele. Essas variações de luz são resumidas e transmitidas para o computador [2].

A aplicação das luvas de dados se dá em diferentes atividades, como na indústria da animação para dar movimento a personagens e na medicina por meio da simulação de procedimentos cirúrgicos em ambientes virtuais [3]. A luva de dados 5DT utiliza a tecnologia de fibras óticas para capturar os movimentos dos dedos. Esta categoria de luva possui modelos com 5 e 14 sensores e foi utilizada no processo de integração e suporte ao *framework*.

III. CYBERMED

O CyberMed é um *framework* baseado em RV cujo objetivo é permitir o desenvolvimento de aplicações voltadas ao ensino e treinamento médico através de explorações interativas do corpo humano e da simulação realista de procedimentos médicos em ambientes virtuais imersivos [4, 5]. Ele apresenta uma série de funcionalidades e suporte a diferentes tipos de dispositivos que podem ser facilmente integrados a aplicações a partir de uma interface em alto nível. Desta forma o usuário desta ferramenta não necessita de conhecimentos profundos sobre o dispositivo ou metodologias e algoritmos específicos, uma vez que os detalhes são abstraídos pelo CyberMed.

Embora tenha sido desenvolvido de modo a permitir uma programação de alto nível, o modo de concepção do CyberMed permite expandir suas funcionalidades através de uma programação de mais baixo nível [6]. Este *framework* foi desenvolvido utilizando padrões de projeto de *software* que tornam possível a fácil expansão e integração de novas funcionalidades, assim como o suporte a novos dispositivos.

IV. INTEGRAÇÃO DA LUVA DE DADOS AO FRAMEWORK

O *framework* CyberMed possui uma estrutura de hierarquia para o gerenciamento dos dispositivos e qualquer novo dispositivo integrado deve segui-la. No topo desta hierarquia existe uma classe que todos os dispositivos herdam (CybDevice) e que declara algumas funções abstratas

relacionadas ao funcionamento de qualquer dispositivo, como liga e desliga, por exemplo. Abaixo dessa, tem-se uma classe mais geral que é uma interface em comum para um tipo de dispositivo, por exemplo, para a luva de dados é preciso criar uma classe que contenha funções comuns a qualquer tipo de luva de dados. No final da hierarquia, tem-se a classe do dispositivo real, que deve implementar as funções abstratas de sua superclasse (Figura 1).

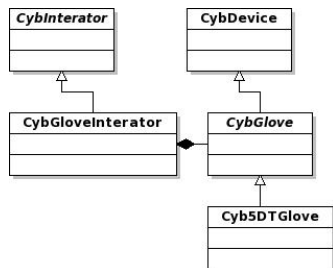


Figura 1. Diagrama de classes simplificado.

Assim, para oferecer suporte a luva de dados, criou-se primeiramente uma classe abstrata que servisse de superclasse para qualquer tipo de luva de dados e em seguida foi desenvolvida a classe da luva de dados, utilizada neste projeto, como subclasse desta. Estas classes são CybGlove e a Cyb5DTGlove (Figura 1).

A classe CybGlove foi desenvolvida buscando conter as tarefas mais gerais que uma luva de dados poderia apresentar (como captura da posição de cada sensor, liga e desliga, por exemplo), criando uma interface em comum para qualquer tipo de luva de dados. Para a implementação da Cyb5DTGlove, criou-se uma classe a partir das funções oferecidas pela luva de dados, além da sobrescrita das funções abstratas especificadas por sua superclasse.

Quando um dispositivo é utilizado no desenvolvimento de uma aplicação, geralmente sua representação visual é relacionada a um modelo tridimensional. Por exemplo, um dispositivo háptico pode ser representado por um modelo de bisturi na sua visualização. No caso de uma luva de dados, entretanto, o modelo normalmente utilizado é o de uma mão. Para que ocorra a interação entre o dispositivo e o modelo tridimensional que o representa na aplicação, tem-se mais uma relação de herança. A CybInterator é uma classe do *framework* responsável por esta interação, mas é preciso que se crie uma subclasse sua para cada tipo de dispositivo. Esta classe armazena o modelo que será utilizado pelo dispositivo enquanto que sua subclasse armazena uma referência ao tipo de dispositivo. No caso da luva de dados, foi desenvolvida a classe CybGloveInterator que contém um objeto do tipo CybGlove e a sua superclasse mantém a referência ao modelo da mão que será utilizado. Porém para que fosse possível armazenar o

modelo da mão na CybInterator foi preciso criar uma nova abordagem para gerenciar tal modelo, o que será discutida na próxima seção.

A. CybModel – suporte à modelos compostos

Durante o planejamento das novas classes constatou-se uma dificuldade, que foi o fato de que os dispositivos até então implementados no *framework* só podiam ser relacionados a um único modelo. Porém, o modelo da mão que seria usado pela luva de dados necessita de vários modelos deste tipo, pois cada parte da mão (dedo) responde a um sensor do dispositivo. Para resolver este problema desenvolveu-se um conjunto de classes que funciona internamente como um novo tipo de modelo composto (CybModel). Esta nova classe é composta por outras duas, a CybMultiModel que gerencia as partes do modelo composto e a CybUpdateModel, que é responsável por indicar como o tipo de dispositivo conectado deve atualizar o modelo, visto que pode-se ter tipos de luvas com diferentes quantidades de sensores (Figura 2).

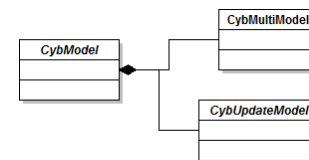


Figura 2. CybModel – Diagrama de classes simplificado.

A CybModel é uma classe abstrata, e as classes que herdam desta é que devem especificar como o modelo vai ser desenhado indicando o posicionamento das partes que o compõe. O CybUpdateModel também é uma classe abstrata e para cada tipo diferente de dispositivo que interage com o modelo deve ser criada uma subclasse desta. Quem fica responsável por instanciar o CybUpdateModel adequado ao dispositivo é a subclasse da CybModel, que deve implementar um método *factory* (*setUpdateModel*) definido como abstrato em sua superclasse. Este método recebe uma *string* com o nome do dispositivo conectado, que é fornecido internamente pelo *framework* e, a partir deste parâmetro, deve retornar a instância adequada.

A classe CybModel foi estruturada de modo que pudesse ser utilizada com qualquer dispositivo que necessite de modelos constituídos por várias partes, como o modelo da mão utilizado pela luva de dados. Um objeto deste tipo foi adicionado ao CybInterator, permitindo assim o armazenamento do modelo da mão.

B. Modificações no CyberMed

Devido à estrutura do CyberMed, pode-se expandir suas capacidades sem alterar códigos já existentes. Por isso, no que diz respeito ao suporte da luva de dados, as mudanças

que ocorreram foram modificações que não alteraram suas classes superiores. Porém, devido ao fato da luva de dados trabalhar de forma diferente dos outros dispositivos na interação, com o modelo composto, foi preciso oferecer suporte a classe CybModel. No desenvolvimento do suporte a esta classe, procurou-se modificar o mínimo possível o código já existente e também houve a preocupação de fazer com que aplicações previamente desenvolvidas não ficassem incompatíveis com a nova versão do *framework*. As alterações consistiram na sobrecarga de funções, o acréscimo de novos campos e de funções para gerenciar o CybModel. Ao final, testes foram realizados para comprovar que a versão era compatível com as aplicações já desenvolvidas.

V. APLICAÇÕES E RESULTADOS

Para testar a efetividade do suporte a luva de dados, foi criada uma aplicação utilizando as novas funcionalidades implementadas. Antes, entretanto, foi preciso definir como o modelo da mão funcionaria.

Para a interação com a luva de dados utilizou-se o modelo de uma mão tridimensional. O modelo foi dividido em várias partes e acrescentaram-se pequenas esferas entre as articulações para proporcionar um melhor efeito visual durante a rotação dos dedos. Em seguida, cada parte do modelo foi exportada separadamente para que fosse possível sua manipulação de forma adequada dentro da aplicação, pois é necessário fazer com que os dedos se movimentem de forma independente em relação aos outros e também de forma hierárquica entre si. Para cada dedo tem-se a seguinte hierarquia de movimentos:

base do dedo → meio do dedo → ponta do dedo

ou seja, a parte do dedo mais à esquerda desta sequência afeta todas as outras a sua direita. Logo, se movimentarmos a base do dedo, moveremos também o meio e a ponta do dedo, se movimentarmos o meio do dedo, moveremos também sua ponta e se movimentarmos a ponta do dedo, apenas está será afetada por ser a última parte da hierarquia. Para obedecer esta hierarquia deixou-se que as transformações causadas no espaço que afetam a base do dedo também sejam utilizadas para afetar o espaço das outras partes que o constituem, ou seja, as articulações abaixo na hierarquia sofrem as transformações no espaço causadas pelas articulações acima desta.

Para utilizar o modelo da mão na aplicação, foi preciso utilizar o CybModel, pois se trata de um modelo composto. Assim, para a aplicação desenvolveu-se uma subclasse da CybModel, a CybHandModel, e a partir dos recursos oferecidos em sua superclasse fez-se a carga e o posicionamento de cada parte da mão. O próximo passo foi criar uma subclasse da CybUpdateModel, para indicar como as informações vindas da luva de dados deveriam afetar o modelo. O modelo da mão possui 18 pontos de

articulação enquanto a luva utilizada para os testes, a *5DT Data Glove 14 Ultra*, possui 14 sensores. Portanto, para a articulação da ponta do dedo utilizou-se 14 valores da articulação. Na Figura 3 tem-se um diagrama de classes simplificado com o relacionamento destas classes e na Figura 4 o resultado da implementação da aplicação.

Foram desenvolvidas, ainda, mais duas aplicações: a primeira delas consistiu na integração da luva de dados com um *tracker* e a segunda em uma aplicação colaborativa entre dois usuários, cada um utilizando uma luva.

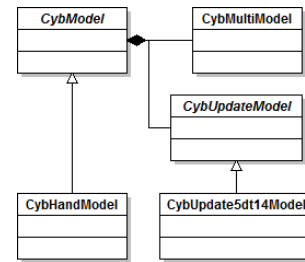


Figura 3. Diagrama de classes simplificado da CybModel para a aplicação.

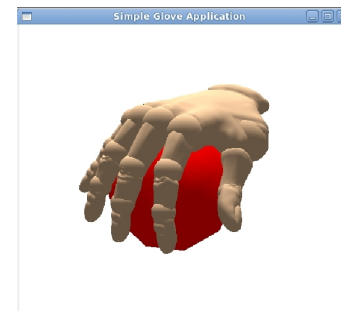


Figura 4. Aplicação desenvolvida utilizando a classe do CyberMed para suporte a luva de dados.

A aplicação com integração da luva com o *tracker* deu-se de forma bastante natural devido à própria estrutura do *framework*. Para o desenvolvimento desta aplicação não foi preciso fazer nenhuma alteração no CyberMed, pois o mesmo permite utilizar os dois dispositivos conjuntamente e também oferece um fácil acesso a ambos. Nesta aplicação criou-se uma nova versão para a CybHandModel e para CybUpdate5dt14Model da aplicação anterior, com o propósito de também utilizar as informações vindas do *tracker* para atualizar a posição do modelo da mão. Assim, além dos movimentos dos dedos, os movimentos da mão também puderam ser visualizados no ambiente virtual (Figura 5).

Para o desenvolvimento da aplicação colaborativa utilizou-se uma segunda luva de dados, da mesma família de luvas porém com 5 sensores (*5DT Data Glove 5 Ultra*). Neste caso, foi preciso especificar como esta nova luva iria atualizar o modelo da mão utilizando os 5 sensores. Entretanto, foi necessário apenas criar uma nova subclasse da

CybUpdateModel (Figura 6). Por apresentar 5 sensores, foi preciso reutilizar o valor de cada sensor em mais de um ponto de articulação do modelo da mão. Assim para o mesmo modelo tem-se duas formas de atualizá-lo dependendo da luva conectada.



Figura 5. Aplicação utilizando uma luva de dados e um tracker.

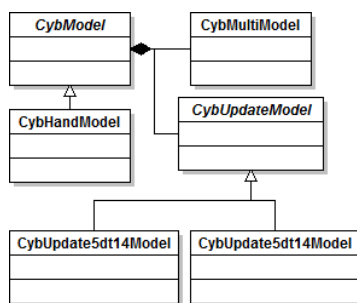


Figura 6. Diagrama de classes simplificado da CybModel para a aplicação colaborativa utilizando dois modelos de luva de dados.



Figura 7. Aplicação colaborativa com 2 luvas de dados.

O CyberMed já possui um módulo de colaboração, o CybCollaboration, que permite que todos os serviços já disponíveis no *framework* sejam utilizados também para a construção de simulações médicas colaborativas [7]. Utilizando este módulo, foram desenvolvidas duas novas classes que dão suporte às luvas de dados utilizadas, permitindo seu uso em aplicações colaborativas. A aplicação consistiu de duas mãos, dispostas uma ao lado da outra, onde a mão do lado esquerdo era a mão do usuário local e a mão do lado direito era a mão do usuário remoto (Figura 7).

VI. CONSIDERAÇÕES FINAIS

A luva de dados proporciona uma forma de interação bastante intuitiva, visto que a manipulação de objetos com as mãos é predominante nos seres humanos. A integração deste dispositivo com o *tracker* proporciona uma interação ainda mais realista, uma vez que permite detectar não apenas os movimentos dos dedos do usuário, mas também a posição de sua mão no ambiente virtual.

A utilização conjunta desses dispositivos no *framework* evidencia a efetividade do suporte implementado à luva de dados, de modo que os dois dispositivos podem ser utilizados conjuntamente. Esta integração ocorreu segundo os padrões de projeto do *framework*, de maneira que a utilização de luvas de dados em uma aplicação colaborativa ocorreu de modo similar à utilização de outros dispositivos presentes no *framework* neste tipo de aplicação.

Como requisito ao suporte a luvas de dados, o presente trabalho também envolveu a inclusão de suporte a modelos compostos. Neste caso, foi necessário realizar o tratamento hierarquizado de modelos de modo a fazer com que as diversas partes da mão se movessem corretamente.

Devido ao fato de o CyberMed conter uma interface em comum para qualquer tipo de luva de dados, tem-se a vantagem de poder programar sem se preocupar com o modelo de luva que será utilizado na aplicação. Portanto, as aplicações desenvolvidas utilizando esta classe mais geral irão funcionar com qualquer tipo de luva integrada ao *framework*.

AGRADECIMENTOS

Este trabalho é apoiado pelo CNPq / INCT-MACC (Processos 573710/2008-2, 183234/2011-1 e 312375/2009-3). Os autores agradecem à professora Fátima Nunes pelo empréstimo da luva com 5 sensores utilizadas nos testes.

REFERÊNCIAS

- [1] C. Kirner and R. Siscoutto. "Fundamentos de Realidade Virtual e Aumentada," in *Realidade Virtual e Aumentada : Conceitos, Projetos e Aplicações*, Kirner e Siscoutto, 2007, 1st ed., SBC, pp. 2-21.
- [2] A. V. Netto, L. S. Machado, "Realidade Virtual: Definições, Dispositivos e Aplicações," 2007, notas didáticas, ICMC-USP.
- [3] R. Haluck, R. Webster et. al. (2011) Tracking Laparoscopic Surgical Motions Using the 5DT DataGlove. [Online]. Available: <http://cs.millersville.edu/~webster/haptics/handproject/handproject.html>
- [4] L.S. Machado, S.F. Campos, Í. L. L. Cunha, R. M. Moraes, "CyberMed: Realidade Virtual para Ensino Médico," in IFMBE Proceedings, 2004, pp. 573-576.
- [5] A. Carvalho Jr, D. Souza, L.Machado, "Utilização de Rastreadores Magnéticos no Desenvolvimento de Aplicações com Realidade Virtual para a Educação," in *Anais do WRVA*, 2009.
- [6] R.M. Moraes, L.S. Machado. "Development of a Medical Training System with Integration of Users' Assessment". In: Jae-Jin Kim. (Org.) *Virtual Reality*. : Intech, 2011, pp. 325-348.
- [7] B. Sales, L. Machado, R. Moraes, "Colaboração Interativa para Sistemas de Realidade Virtual voltados ao Ensino e Treinamento Médico," in *Cong. Brasileiro de Inf. em Saúde (CBIS 2010)*, 2010.