

Programação Visual para um *Framework* de Desenvolvimento de Aplicações Médicas Baseadas em Realidade Virtual

Aline Marques de Moraes¹, Liliane dos Santos Machado²

^{1,2}Laboratório de Tecnologias para o Ensino Virtual e Estatística (LabTEVE), Departamento de Informática, Universidade Federal da Paraíba (UFPB), Brasil.

Resumo - Aplicações computacionais baseadas em Realidade Virtual (RV) tem se destacado em tarefas de treinamento e ensino na área médica devido à sua capacidade de simular ambientes e cenários realistas, interativos e imersivos. Estas simulações permitem aos usuários treinar habilidades e a tomada de decisão diante de situações diversas. A complexidade envolvida no processo de geração de aplicações médicas baseadas em RV exige a manipulação de ferramentas computacionais. Neste contexto, se destaca o CyberMed como *framework* para desenvolvimento de aplicações médicas baseadas em RV. Observa-se que tanto o CyberMed quanto outros *frameworks* com as mesmas potencialidades não possuem uma interface voltada para leigos em programação de aplicações, como os profissionais de saúde. Esse artigo apresenta um estudo relacionado a estes *frameworks*, com o objetivo de analisar suas características e verificar o nível de conhecimento necessário para utilizá-los. A partir dessa pesquisa será proposta a criação de um sistema de programação visual integrado ao CyberMed para facilitar a criação de aplicações baseadas em RV para a área médica.

Palavras-chave: programação visual, realidade virtual, *framework*, medicina.

Abstract - Computer applications based on Virtual Reality (VR) has been outstanding in training and teaching jobs in the medical field because of its ability to simulate realistic scenarios and environments, interactive and immersive. These simulations allow users to practice skills and decision making in different areas. The complexity involved in the generation of medical applications based RV requires manipulation of computational tools. In this context, CyberMed is a framework for developing medical applications based on VR for medical purposes. It was observed that both the CyberMed as other frameworks with the same capabilities do not have an interface easy to be used by health professionals. This article presents a study related to these frameworks, in order to analyze their characteristics and check the level of knowledge needed to use them. From this research is proposed a visual programming system integrated to CyberMed to facilitate the development of applications based on VR for the medical field.

Keywords: Visual programming, virtual reality, medicine.

Introdução

Uma técnica de programação, que surgiu na década de 60 (1), chamada programação visual, utiliza a eficiência dos métodos de visualização para tornar a interação do usuário com sistemas complexos uma atividade simples. O amadurecimento da programação visual tornou essa técnica essencial em sistemas de desenvolvimento rápido (2), ferramentas que envolvem áreas distintas da computação (3) e sistemas de execuções complexas (4). Vale ressaltar que uma nova vertente de utilização da programação visual vem sendo proposta para os ambientes de Realidade Virtual (RV) (5). Uma dessas propostas é a integração da programação visual com sistemas para a geração de

aplicações médicas baseadas em RV.

A RV é uma área de pesquisa que reúne conhecimentos de vários segmentos como a eletrônica, computação, robótica, física, psicologia, dentre outras(6). O objetivo da RV é oferecer sistemas em tempo-real que integrem aspectos de imersão e interatividade para simular ambientes realistas. Para isso são utilizados equipamentos específicos que estimulam os sentidos, explorando os sistemas visual, tátil, auditivo e olfativo humano.

É observado que a maioria dos *frameworks* que utiliza elementos de RV, em sua concepção ou uso, apresenta uma facilidade de manipulação restrita, o que os limita a usuários com conhecimento específico em RV. Isso é causado, principalmente, porque o foco desse tipo de

framework é oferecer um maior realismo para as aplicações em RV geradas, livrando o programador de detalhes das técnicas necessárias. Entretanto, estes não consideram o público-alvo leigo em programação. Essa problemática pode ser observada em *frameworks* baseados em RV voltados para treinamento e estudo na área médica, que exigem conhecimento de programação pelo público-alvo: os profissionais de saúde. Por isso, técnicas e estudos relacionados a melhoria na interação do usuário com esses *frameworks* são necessárias para a utilização dessas ferramentas pela comunidade em larga escala.

Esse artigo irá analisar os *frameworks* baseados em RV e destinados à área médica que se utiliza de algum tipo de recurso visual. Feito o estudo sobre as características desses *frameworks*, será proposta uma alternativa de integração entre programação visual com um *framework* de geração de aplicações médicas baseadas em RV. Nesse estudo será o CyberMed, através de um sistema de programação visual. Tal sistema servirá como ponte de comunicação entre o usuário e o *framework* para facilitar o processo de criação de aplicações em RV.

Programação Visual

A programação visual é comumente definida como o uso de expressões visuais (tais como gráficos, desenhos, animação ou ícones) no processo de programação (7). A definição de programação visual engloba os seguintes componentes que podem ser vistos na Figura 1: linguagens de programação visual (LPV), expressões de programação visual (EPV), sistemas de programação visual (SPV) e ambientes de programação visual (APV)(8).



Figura 1. Diagrama dos componentes da programação visual (botar as letras)

A programação visual é aquela que utiliza mais de uma dimensão para transmitir um determinado significado(8). Essas dimensões

podem ser objetos, relacionamentos espaciais, ou até mesmo dimensões temporais para especificar as relações semânticas de antes e depois. Cada relacionamento ou objeto multidimensional e potencialmente significativo é chamado de símbolo. A coleção de um ou mais símbolos é vista como uma expressão visual (EPV)(Figura 1a). Essas expressões podem ser diagramas, ícones, objetos gráficos, etc. Quando uma linguagem de programação contém expressões visuais (Figura 1c) ela passa a ser chamada de linguagens de programação visual (LPV).

Uma LPV, também conhecida como programação diagramática, é qualquer linguagem de programação que permite que usuários criem programas de manipulação de elementos do programa graficamente ao invés de especificá-los textualmente(9). Quando expressões visuais são usadas num ambiente de programação como atalho para gerar código de edição que pode ou não ter uma sintaxe diferente daquela usada para editar código (Figura 1b), o ambiente é chamado de ambiente de programação visual (APV). Um sistema de programação visual (SPV) é um sistema para a criação alto nível de aplicações que foram projetadas com LPV(Figura 1d)(10). Um SPV utiliza o que é essencialmente visual ao invés de programação linguística. Geralmente, eles são de natureza pictórica, desde os ícones para representar os dados, até operações. Por sua vez, os APV são aqueles onde a língua é textual, mas o depurador, o verificador, e outros elementos, podem ter elementos visuais e editores de texto visual (e) que suportam programação textual, fazendo uso de técnicas visuais, como o recuo, fontes e cores (11).

Frameworks médicos com recursos visuais

Um *framework* é definido como uma estrutura de suporte no qual outros programas podem ser organizados e desenvolvidos(12). Ele pode incluir suporte a programas, código de bibliotecas, uma linguagem de *scripting*, ou outro *software* para ajudar a desenvolver ou unir diferentes componentes de um projeto de *software*. Particularmente, para o desenvolvimento de aplicações baseadas em RV para a área médica, o uso de *frameworks* tende a facilitar este processo, uma vez que elimina a necessidade do programador conhecer os detalhes de implementação de cada técnica, seja ela de visualização estereoscópica, interação háptica (com sensação de toque), deformação interativa de modelos, etc.

A programação visual pode estar presente em

frameworks de vários segmentos como a estatística(13) e a geologia(14), por exemplo. Mas, uma área que é possível destacar como importante para a integração da PV com um *framework* voltado ao desenvolvimento de aplicações baseadas em RV é a área médica, devido à necessidade de aplicações de apoio à prática e ensino nesta área. Baseado nesse contexto, foi realizado o levantamento de *frameworks* que disponibilizam técnicas de RV voltados ao desenvolvimento de aplicações nesta área que utilizam algum tipo de recurso visual, particularmente técnicas de PV. Foram encontrados três *frameworks* que ofereciam algum tipo de técnica visual de interação e que serão detalhados a seguir.

a) MeVisLab

MeVisLab é um *framework* que fornece uma interface multiplataforma de programação visual modular, com um conjunto abrangente de processamento de imagens e ferramentas de visualização. Ele tem como objetivo propor uma alternativa para otimizar o tempo de desenvolvimento com o pacote ITK. O *Insight Toolkit* (ITK) é um *software open-source* composto por um pacote de ferramentas para programação na linguagem C++ para o registro e execução de segmentação em imagens(4). Este pacote não propicia suporte à visualização e análise, nem oferece qualquer interface gráfica para o usuário. Deste modo, o MeVisLab combina uma série de conceitos de *design* para prototipagem rápida de aplicações voltadas à visualização e análise, fornecendo um conjunto abrangente de processamento geral de imagens e ferramentas de visualização para imagens médicas multidimensionais, bem como métodos avançados de órgãos específicos e as tarefas clínicas(16). Pelas características apresentadas de recursos gráficos baseados em módulos, com presença de linguagem textual e expressões visuais, é possível afirmar que o MeVisLab se enquadra num Ambiente de Programação Visual (APV).

b) ViMeTWizard

O ViMeT (*Virtual Medical Training*) é um *framework* nacional, orientado a objetos, desenvolvido com a finalidade de reaproveitar aplicações codificadas de forma isolada para simular algumas tarefas médicas com realidade virtual. Com o ViMeT, é possível gerar aplicações baseadas em RV, além de utilizar outros recursos

como detecção de colisão, deformações e esteoreoscopia(14). A interação de usuários não especializados com esta ferramenta só ocorreu com desenvolvimento de ViMeTWizard.

O ViMeTWizard é um componente visual do ViMeT que permite a criação de aplicações, alterações de parâmetros ou até mesmo a chamada de aplicações pré-definidas. Todas essas ações ocorrem através de chamadas gráficas em estrutura de janelas, botões e gráficos que geram a aplicação final baseada em RV. Contudo, o ViMeTWizard não oferece todas as funcionalidades que o ViMeT disponibiliza. Nesse contexto, pode-se afirmar que o ViMeTWizard faz uso apenas de recursos visuais, principalmente pelo fato de ser projetado em uma interface visual. Entretanto, seu destaque é relevante pelo fato deste ser um trabalho nacional que buscou melhorar a interação de usuários da saúde leigos na utilização de ferramentas de desenvolvimento com RV.

c) SCIRun

O SCIRun é um *framework* baseado em RV, que não é voltado exclusivamente para a área médica. Ele possui simulações em grande escala que podem ser compostas, executadas, controladas e ajustadas de forma interativa. O SCIRun foi desenvolvido para reduzir o tempo que o cientista gasta no processo de modelagem e para fornecer uma ferramenta para a exploração da ciência computacional e para os problemas de engenharia(3). Além disso, ele permite que o usuário controle interativamente simulações científicas, enquanto a computação está em andamento. O SCIRun usa uma interface de programação visual para permitir que o cientista construa as simulações através de poderosos componentes computacional. Embora o ambiente de programação visual seja o foco central de SCIRun, ele requer um poderoso conjunto de ferramentas computacionais.

A criação da simulação é realizada através de uma interface de programação visual, pois para executar o programa, é necessário especificar os parâmetros de uma interface gráfica do usuário. Dessa forma, pode-se afirmar que o SCIRun é um Ambiente de Programação Visual devido à composição de elementos gráficos como fluxogramas.

Resultado da análise

A partir das características apresentadas a respeito dos *frameworks*, pode-se resumi-las na

Tabela 1.

Tabela 1. Comparação dos Frameworks apresentados

Parâmetros	Possui programação visual?	Interface criada com que intuito?	Código aberto?	Recursos presentes de RV
Frameworks				
MeVisLab	Sim	Propor uma alternativa para otimizar o tempo de desenvolvimento com o pacote ITK	Não	Utilização de aparelhos de medição, com retorno de força (simulação de toque)
ViMeTWizard	Não	Criar e modificar aplicações desenvolvidas com o <i>framework</i> ViMeT	Sim	Simulação de toque (retorno háptico), detecção de colisão com precisão, deformação e estereoscopia
SCIRun	Sim	Fornecer uma ferramenta para a exploração da ciência computacional e para os problemas de engenharia	Sim	Simulação de toque (retorno háptico) e estereoscopia

A Tabela 1 permite observar que não existe um *framework* voltado ao desenvolvimento de aplicações para a área médica com técnicas de RV que apresente uma completa integração entre a programação visual e o *framework*. Dessa forma, ao avaliarmos os parâmetros apresentados, é possível perceber que o SCIRun, embora tenha elementos de programação visual, não foi desenvolvido diretamente para a área de saúde, ou seja, necessidades médicas específicas não serão atendidas por essa ferramenta.

O ViMeTWizard, embora não utilize explicitamente a PV, foi relevante devido à sua proposta de melhoria na forma de uso da ferramenta de desenvolvimento. Finalmente, o MeVisLab possui uma variedade maior de recursos em RV na sua composição, mas é uma ferramenta proprietária, o que torna seu acesso restrito.

Ao analisar as características de cada um dos *frameworks*, é possível notar que estes utilizam técnicas de programação visual de modo limitado, ou seja, em todos eles é necessária a inserção de comandos textuais ou parâmetros para a geração das aplicações, tornando inevitável o uso de linhas de código.

Proposta

Após a análise dos *frameworks* médicos detentores de algum tipo de recurso visual foi possível observar que, embora objetivem tornar a criação de novas aplicações mais rápida e a interação com o usuário leigo mais intuitiva, estes utilizam de forma limitada as técnicas de PV. Além disso, não foi identificado na literatura um

sistema de programação visual (SPV). Neste caso, o SCIRun e o MeVisLab se caracterizam como APV e o ViMeTWizard como uma ferramenta que utiliza recursos visuais, não se enquadrando na definição de fato de PV.

Outro fator que justifica a utilização inadequada de recursos gráficos ou até mesmo PV em *frameworks* para RV pode ser comprovada em uma pesquisa recente(15) que afirma que não existe um padrão único nos *frameworks* dotados de técnicas de RV para desenvolvimento para a saúde. Por isso, cada *framework* analisado possui um processo de desenvolvimento e uma técnica visual de interação aparentemente diferentes. Nessa mesma pesquisa, dentre os *frameworks* relatados, o mais completo em termos de funcionalidades foi CyberMed, cuja disponibilização é livre.

O CyberMed é um *framework* voltado ao desenvolvimento de sistemas de RV baseados em PCs cujo objetivo é oferecer suporte na criação de aplicações que envolvam os conceitos de RV, particularmente aquelas destinadas à simulação de procedimentos médicos em um ambiente virtual imersivo(16). Ele trabalha com um conjunto de bibliotecas, escritas em C++, que visam o desenvolvimento rápido de aplicações em RV. Tais bibliotecas oferecem recursos de visualização monoscópica e estereoscópica, suporte aos dispositivos de visualização 3D, suporte a sistema hápticos, detecção de colisão, deformação interativa, importação automática de modelos VRML, customização de menus, entre outras características. Pelo fato do CyberMed ser um *framework* voltado para a programação de aplicações baseadas em RV, existe uma

dificuldade no uso desse *framework* por pessoas que não tenham um conhecimento específico em computação ou programação. Pelo fato deste *framework* ser voltado a área médica, sua utilidade pode ser estendida para profissionais de saúde, bastando a oferta de uma alternativa visual de interação do CyberMed para esse público-alvo.

Baseado no estudo feito, anteriormente, sobre os recursos visuais dos *frameworks* médicos em RV e utilizando os resultados da pesquisa relatada, é proposta a criação de um Sistema de Programação Visual integrado ao CyberMed para melhorar a utilização do *framework* pelos usuários leigos, especificamente profissionais de saúde. O Sistema de Programação Visual para o CyberMed, chamado de CyberMedVPS, tem como objetivo ser uma alternativa para que os usuários leigos possam utilizar o *framework*, pelo menos nas funcionalidades principais, como visualização e criação de aplicações baseadas em RV. Esse sistema é composto, inicialmente, por um módulo centralizador de visualização integrado com uma interface de comunicação com o CyberMed. O módulo centralizador executa de forma transparente para o usuário, pois essa unidade é desenvolvida baseada em estudos de programação visual. A interface é quem centraliza todas as funcionalidades que já são disponíveis no CyberMed e de outros *frameworks*, pois sua concepção será baseada no princípio de ser genérica para outros *frameworks* baseados em RV. A sua arquitetura pode ser observada na Figura 2:

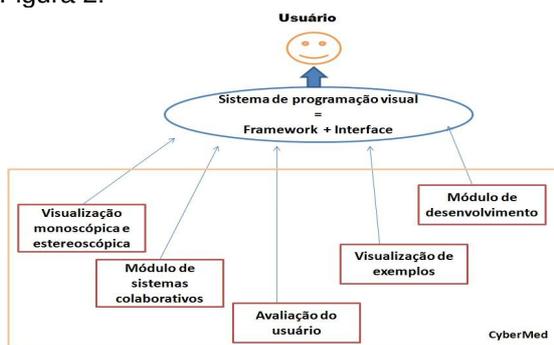


Figura 2. Arquitetura inicial do CyberMedVPS agregado ao CyberMed

A tecnologia a ser utilizada para prover o sistema de programação visual CyberMedVPS é a linguagem Java, devido à sua característica multiplataforma e adequação à programação orientada a objetos do *framework*. As chamadas entre a interface de comunicação e o CyberMed serão possíveis por meio de métodos em JNI ou chamadas nativas para a execução do CyberMed.

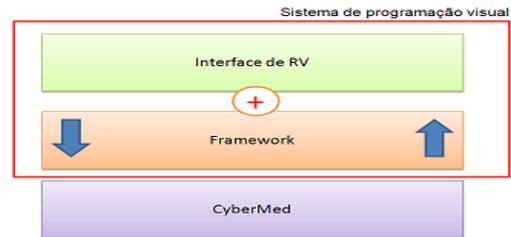


Figura 3. Arquitetura do CyberMedVPS agregado ao CyberMed

A estrutura hierárquica do CyberMedVPS, apresentada na Figura 3, esclarece como é importante ter um elo de comunicação entre a camada de baixo nível, que é representado pelo CyberMed, e o componente gráfico que será para utilização do cliente que é representado pelo *framework*. Ou seja, a interface de integração é a ponte de tradução das chamadas que irão acontecer do *framework* para a camada em baixo nível, no caso a camada referente ao CyberMed; e vice-versa. É importante definir que o CyberMedVPS é a união entre a interface de integração baseada em RV e o *framework* de comunicação com o usuário.

Conclusão

O estudo realizado permitiu analisar as necessidades e desafios relacionados ao provimento de ferramentas para profissionais da área médica para que estes possam desenvolver aplicações baseadas em realidade virtual para ensino e treinamento. Observou-se que para a área de RV poucas ferramentas possuem alternativas que preencham essa lacuna. Embora os profissionais de saúde sejam os principais beneficiados com o desenvolvimento dessas ferramentas médicas, para treinamento ou estudo, praticamente inexistem ferramentas em RV que estes consigam usufruir pelo fato da necessidade de conhecimentos de programação.

Através da análise feita, foi possível inferir que um dos grandes desafios nos *frameworks* baseados em RV e voltados para área médica é oferecer sistemas que permitam desenvolver aplicações que reproduzam aquilo que o profissional de saúde experimenta em situações reais atrelado a mecanismos de programação visual que sejam eficientes para a comunicação com o usuário, principalmente porque os operadores, que no caso do CyberMed são profissionais de saúde e na maioria leigos em computação, não deveriam precisar lidar com código ou arquivos de configuração, mas sim

poder trabalhar com interfaces gráficas intuitivas(18). Com base nesse contexto torna-se relevante desenvolver mecanismos que facilitem a cognição do usuário, como as técnicas de programação visual. A proposta do CyberMedVPS foi apresentada como uma dessas alternativas que permitirá inserir usuários não habituados a manipular *frameworks* dessa natureza.

Referências

1. Goldstine HH. The Computer From Pascal To Von Neumann. Publicado pela Priceton University Press, ISBN: 0-691-08104-2, 1972.
2. Nunes FLS, Oliveira ACM, Rossatto DJ. ViMeWizard: Uma Ferramenta Para Instanciação De Um *Framework* De Realidade Virtual Para Treinamento Médico. Congresso Latinoamericano de Informática - CLEI, 2007.
3. Jonhnson CR e Parker SG. Applications In Computational Medicine Using Scirun: A Computational Steering Programming Environment. Proceedings H.W. Meuer, editor, Supercomputer '95, pages 2–19. Saur-Verlag, 1995.
4. Augustine KE, Holmes III DR e Robb RA. ITK And ANALYZE: A Synergistic Integration. Proceedings SPIE 5367, 6 (2004), DOI:10.1117/12.533484, 2004 .
5. Paquet J e Kropf P. The GIPSY Architecture".Proceedings of Distributed Computing on the Web, Quebec City, Canada, 2000.
6. Roseblum L, Burdea G e Tachi. VRReborn, CG&Applications, 18(6):21-23, 1998.
7. Paquet J, Plaice J. Dimensions And Functions As Values. Proceedings of the Eleventh International Symposium on Lucid and Intensional Programming, Sun Microsystems, Palo Alto, California, EUA, 1998.
8. Burnett MM e Mcintyre DW. Visual programming. IEEE Computer, 28(3):10-19, March 1995. Special issue on visual programming.
9. Johnston WM, Hanna JRP e Millar RJ. Advances In Dataflow Programming Languages(PDF). ACM Computing Surveys 36 (1): 1-34. doi:10.1145/1013208.1013209, 2004. 10. <http://www.msri.org/about/sgp/jim/software/vps/de-sc/node2.html>, Acessado em maio de 2010.
11. Nardi BA. A Small Matter Of Programming: Perspectives On End User Computing. Publicado pela TechBooks. ISBN: 0-262-14053-5, 1993.
12. Rahman MM. An Application Of Prototyping Models For Web Application Development Using Php Framework Cakephp, Independent University, Bangladesh, 2007.
13. Gomes RGS, Moraes RM, Machado LS. The Calculadora Estatística for Present and Distance Learning- Proceedings of the 7th International Conference on Teaching Statistics (ICOTS-7), Julho, Salvador, Brasil, 2006.
14. Nunes FLS, Oliveira ACM, Rossatto DJ. ViMeTWizard: Uma Ferramenta Para Instanciação De Um Framework De Realidade Virtual Para Treinamento Médico. Congresso Latino-americano de Informática - CLEI, 2007.
15. Santos AD, Machado LS. Realidade Virtual Aplicada ao Ensino de Medicina: Taxonomia, Desafios e Resultados. In: Anais do Workshop de Realidade Virtual e Aumentada, Santos, Brazil. CDROM, 2009.
16. Rexilius J, Spindler W, Jomier J, K'onig M, Hahn HK, Link F e Peitgen, H. , 2004.
17. Machado LS, Souza DFL, Souza LC, Moraes RM. Desenvolvimento Rápido de Aplicações de Realidade Virtual e Aumentada Utilizando Software Livre. Book Chapter. In: Veronica Teichrieb; Fátima Nunes; Liliane Machado; Romero Tori. (Org.) *Realidade Virtual e Aumentada na Prática*. Porto Alegre: SBC, p. 5-33, 2008.
18. Bastos TA, Silva RJM, Raposo AB, Gattass M. ViRAL: Um Framework para o Desenvolvimento de Aplicações de Realidade Virtual. VII Sympomsium on Virtual Reality, p. 51-62, São Paulo, SP – Brasil, 2004.