

# Um Framework para Interação Gestual com Aplicações de Realidade Virtual via Dispositivos de Interação Natural

Marco Santos Souza<sup>1</sup>, Tiago de Holanda Cunha Nobrega<sup>1 2</sup>, Aldo von Wangenheim<sup>1</sup>

<sup>1</sup>Instituto Nacional para Convergência Digital – Universidade Federal de Santa Catarina

<sup>2</sup>Universidade do Vale do Itajaí (UNIVALI)

sms.cpp@gmail.com, tigarmino@gmail.com, awangenh@inf.ufsc.br

**Abstract.** *This paper presents a framework for developing virtual reality applications (such as games). Among other features, the framework provides developers easy access to the position of the hands of the end user in a three-dimensional space, so that this information can be used to generate interaction with a virtual environment.*

## 1. Introdução

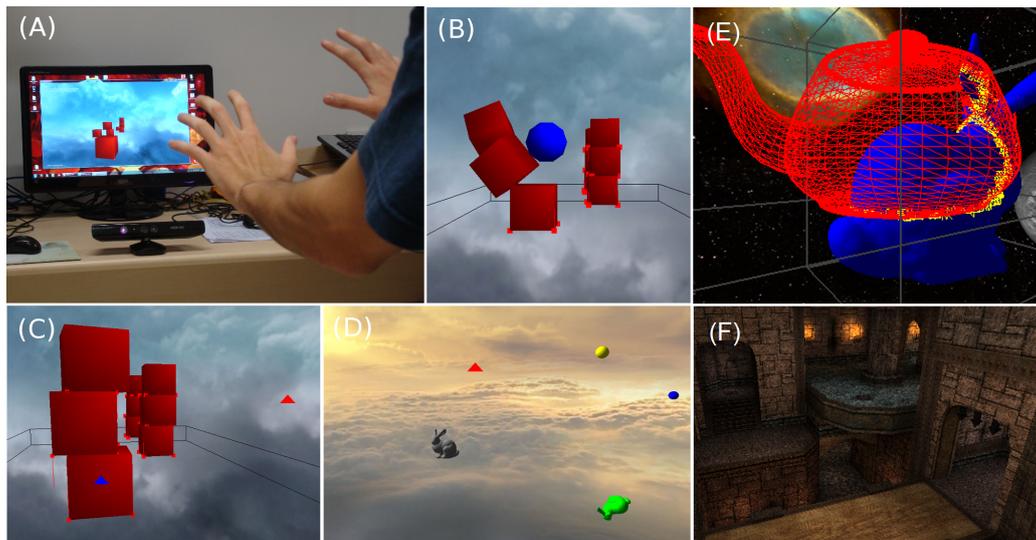
A história das interfaces de interação homem-máquina é um caminho que vai da complexidade à simplicidade [Valli 2004]. Acredita-se que, idealmente, os computadores devem ser capazes de enxergar e processar ações dos usuários sem exigir o uso de algum acessório complicado. Fazer com que os computadores possam “enxergar o mundo” é o objetivo da linha de pesquisa conhecida como *visão computacional* [Marr 1983]. Interação Natural (*Natural Interaction*, NI), é o conceito que diz respeito a interação do ser humano com dispositivos eletrônicos através de gestos, movimentos, expressões corporais e comandos de voz, por exemplo.

A *Open Natural Interaction*, ou *OpenNI*, é uma organização sem fins lucrativos criada em novembro de 2010 com o propósito de certificar e melhorar a interoperabilidade dos diferentes dispositivos de interação natural e das aplicações que usam tais dispositivos [OpenNI 2011]. A API de código aberto disponibilizada pela OpenNI tem o propósito de tornar-se um padrão para comunicação com dispositivos de interação natural.

## 2. O Framework

O framework apresentado neste trabalho oferece uma série de ferramentas e abstrações básicas para agilizar o desenvolvimento de aplicações gráficas, tais como grafo de cena, controle de câmera, *skybox*, *billboards*, carregador de modelos 3D e texturas. Fornece também ferramentas para calcular, salvar e carregar malhas de polígonos representadas com *BVHs* (*bounding volume hierarchy*) – estrutura de dados que pode ser usada para agilizar testes de intersecção entre os polígonos de duas malhas possivelmente complexas (figura 1-E) – e abstrações para carregar e renderizar cenários no formato *BSP*<sup>1</sup> (figura 1-F). Além disso, o framework provê acesso as funcionalidades de uma *engine* para simulação física de corpos rígidos chamada *Planck*, a qual foi desenvolvida em trabalhos anteriores [Souza et al. 2011]. Ainda, além das ferramentas para ler informações de dispositivos de entrada tradicionais como mouse e teclado, o framework implementa

<sup>1</sup>O framework suporta atualmente o formato BSP 46, que é a versão utilizada em jogos como *Quake 3* e *OpenArena*



**Figure 1. Imagens de protótipos desenvolvidos usando o framework proposto.**

abstrações que encapsulam chamadas à API OpenNI, de forma que para o programador-usuário é facilmente possível reconhecer e tratar gestos do usuário final e/ou saber a posição de suas mãos num espaço tridimensional quando um dispositivo de interação natural – como o *Kinect*, da *Microsoft* – estiver disponível.

A figura 1 mostra alguns dos protótipos que foram desenvolvidos com o framework proposto e testados usando o Kinect (1-A). No protótipo ilustrado em 1-B, uma esfera tem sua posição alterada para seguir a mão do usuário, podendo colidir com outros objetos na cena. No mesmo protótipo, usando as duas mãos, pode-se rotacionar e transladar a câmera virtual (1-C). Em outro protótipo, é possível lançar esferas com as mãos para acertar objetos que aparecem na tela (1-D).

O framework faz uso de uma série de *design patterns* de sistemas orientados a objetos e foi implementado em linguagem C++. Entre as poucas dependências externas que possui, adota somente soluções multiplataforma e de código aberto, como o *OpenGL* e a *libjpeg* (para tarefas de renderização e carregamento de imagens, respectivamente). Em sua versão atual, o framework pode ser facilmente compilado para sistemas Windows, Linux e Mac. Para interagir com o Kinect, o OpenNI necessita que algum *middleware* para tal funcionalidade esteja instalado no sistema. Nos protótipos desenvolvidos, utilizou-se o *NITE* – middleware de código aberto da *PrimeSense*, que é a empresa responsável também pelos drivers oficiais do Kinect para o console *Xbox 360*.

## References

- Marr, D. (1983). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. Henry Holt & Company.
- OpenNI (2011). Openni programmer guide.
- Souza, M. S., de H. C. Nobrega, T., Silva, A. F. B., Carvalho, D. D. B., , and von Wangenheim, A. (2011). A rigid body physics engine for interactive applications. *SBGames 2011*.
- Valli, A. (2004). Natural interaction white paper.