

# Interface para Joystick

## Autores

---

Marcelo Antonio Ribeiro Camargo

## Orientador

---

Luis Augusto Consularo

## 1. Introdução

---

Para um melhor entendimento sobre o projeto, esta seção introdutória será destinada à apresentação do problema que originou a necessidade do desenvolvimento. Após uma visão geral sobre as características principais, maiores detalhes e informações poderão ser encontrados abaixo, nos tópicos específicos sobre cada fase do projeto.

Os ambientes de realidade virtual existentes atualmente, mais comuns e de fácil acesso ao usuário final, geralmente são distribuídos para utilização através de um navegador (browser) juntamente com um componente adequado (plug-in). Um exemplo disso é a combinação entre o navegador Internet Explorer da Microsoft, e o componente Cortona da empresa ParallelGraphics. A criação dos ambientes, objetos e animações fica por conta da linguagem de marcação VRML. Existem muitos outros navegadores, componentes e linguagens disponíveis para esta finalidade, entretanto, utilizarei o citado acima como base para este projeto.

Mesmo possuindo ótimos recursos para criação de cenas e uma boa navegabilidade, o ambiente de realidade virtual proposto não oferece uma interação adequada entre o usuário e os objetos, principalmente quando da utilização de dispositivos externos, como joysticks, luvas (gloves) e outros periféricos. Assim, a opção disponível fica a cargo dos periféricos padrões, como o mouse e o próprio teclado, restringindo a interação a apenas duas dimensões.

Uma das soluções para este problema é o uso de uma interface de comunicação entre o componente VRML e os dispositivos externos.

Através do novo esquema proposto, o usuário tem acesso ao ambiente virtual sem passar necessariamente pelo navegador, tendo-o apenas como fonte para saída de dados e visualização. Mantendo-se a comunicação convencional através do mouse e do teclado, a interação torna-se muito mais rica, maleável e abrangente.

Entretanto, somente esta nova interface não é suficiente para a comunicação direta entre o usuário e o ambiente VRML, bem como com os objetos presentes na cena. Para isso, torna-se necessária a utilização de scripts para comunicação interna, totalmente transparentes ao usuário final, porém, completamente configuráveis pelo desenvolvedor.

## 2. Objetivos

---

Como já apresentado na seção anterior, o principal objetivo deste projeto é a criação de uma interface de comunicação entre dispositivos externos e ambientes virtuais, possibilitando assim uma melhor navegabilidade e interação entre o usuário e os objetos presentes na cena.

Atualmente já é possível utilizar o joystick como instrumento de entrada de dados em ambientes VRML, entretanto, os comandos são restritos apenas à navegação e visualização, limitando as opções de acesso ao mundo tridimensional. E ainda, mesmo tratando-se de navegação, as instruções atuais possibilitam apenas dois ou três graus de liberdade (DOF) simultâneos. Este projeto busca também possibilitar a utilização de seis graus de liberdade (6DOF), tanto para navegação quanto para interação com objetos.

Finalmente, o grande desafio e a fonte complexidade não estão voltadas somente para os ambientes e comandos VRML, mas também, para a interface de comunicação, os scripts, arquivos intermediários, e toda a lógica de troca de dados entre o usuário e mundo virtual.

O termo “graus de liberdade” citado acima, refere-se a tipos de movimentação em um determinado espaço, real ou virtual. Normalmente é referenciado pela sigla DOF, acrônimo do inglês “Degrees Of Freedom”. Entre eles, podemos citar:

Translação (deslocamento) no eixo X (esquerda/direita) - swaying    Translação (deslocamento) no eixo Y (cima/baixo) - heaving    Translação (deslocamento) no eixo Z (frente/trás) - surging    Rotação no eixo X - yawing    Rotação no eixo Y - pitching    Rotação no eixo Z - rolling

Existem trabalhos semelhantes a este projeto, alguns deles exatamente com o mesmo objetivo. Antes do início da fase de desenvolvimento de uma interface própria, foram testadas algumas das opções já existentes.

Um exemplo é o Oh\_Joy da empresa GeoVRML. Num primeiro momento, esta solução mostrou-se suficiente para as possibilidades de comunicação entre um dispositivo externo e um ambiente VRML. Todavia, no decorrer dos testes realizados, algumas falhas e limitações foram percebidas, o que acabou gerando a necessidade de criação de uma nova interface.

Assim, quando um recurso possui 6DOF, diz-se que ele tem 6 graus de liberdade e, portanto, a possibilidade de realização de todos os movimentos possíveis, tanto de deslocamento quanto de rotação, bem como, todas as combinações entre eles.

## 3. Desenvolvimento

---

O projeto é composto por vários arquivos, sendo scripts e PROTOS utilizando linguagem VRML e programas desenvolvidos em Visual Basic versão 6.

### Escolha da linguagem

A escolha desta linguagem de programação deve-se ao fato de ser a de maior domínio deste programador no momento, possibilitando assim maior rapidez nas fases de desenvolvimento, testes e correções. Além do VB, qualquer outra linguagem de programação poderia ter sido utilizada, desde que ofereça acesso aos recursos da biblioteca DirectX da Microsoft, versões 7 ou 8, mais especificamente ao DirectInput, responsável pela captação das instruções enviadas pelos dispositivos externos.

## **Biblioteca DirectX**

O Microsoft DirectX é um pacote avançado de APIs (application programming interfaces - interfaces de programação de aplicativos) de multimídia, embutido nos sistemas operacionais Microsoft Windows. O DirectX fornece uma plataforma padrão de desenvolvimento para PCs com Windows, permitindo que desenvolvedores de software acessem características especializadas de hardware sem ter que escrever códigos específicos para hardware. O DirectX surgiu em 1995 e é um padrão reconhecido para desenvolvimento de aplicações multimídia na plataforma Windows.

Simplificando, o DirectX permite que jogos sejam executados no Windows 95/98 e Windows Me e outros. Todos os jogos novos, da maioria dos fabricantes, usaram o DirectX para permitir que seus jogos sejam executados, corretamente, no Windows 95/98 e Windows Me. Também fornece aos desenvolvedores, ferramentas que ajudam a obter a melhor performance possível da máquina que você estiver usando. Automaticamente, determina as capacidades de hardware de seu computador e ajusta os parâmetros do aplicativo de acordo com essas capacidades. Com o DirectX, pode-se até mesmo executar aplicações multimídia que exijam suporte para características que um sistema não possui, simulando certos dispositivos de hardware através de uma "camada de emulação de hardware", que fornecem drivers de software que agem como hardware. Por exemplo, um jogo DirectX, que faz uso de imagens 3D, pode ser executado em uma máquina que não tem uma placa aceleradora, porque o DirectX simula os serviços de uma placa 3D.

Especificamente para este projeto, a utilização do DirectX restringe-se exclusivamente aos recursos do DirectInput.

## **Interface DirectInput**

Resumidamente, esta aplicação realiza uma constante varredura nas instruções de entradas, verifica as eventuais alterações e gera um arquivo em formato WRL, com informações sobre eixos e botões pressionados. Este arquivo transfere os dados para uma PROTO que por sua vez envia os valores para um script presente no ambiente virtual.

## **Arquivo de comunicação**

Para propiciar a comunicação entre a interface programada em VB6, um arquivo em formato VRML é gerado com os dados mais recentes, provenientes do DirectInput. Em tempo real, este arquivo é importado constantemente pela PROTO a cada ciclo de um Timer com valor configurável, e enviado ao script presente no ambiente 3D. Se forem detectadas alterações nos valores das variáveis, as funções responsáveis por cada atributo realizam as atualizações necessárias no mundo virtual.

Esta é apenas uma das formas de comunicação entre o componente VRML e scripts externos. Outra opção bastante utilizada é o EAI (External Authoring Interface), geralmente em conjunto com a linguagem Java.

Este modelo suporta o envio de dados provenientes de joysticks com comando duplo (dual control), POV (point of view) e 12 botões, semelhantes aos utilizados no Playstation II, da Sony.

## **PROTO intermediária**

A prototipação, em VMRL, é o mecanismo que permite que o conjunto original de tipos de nós seja estendido a partir de um arquivo. A construção dos protótipos é muito semelhante às definições de classes em programação orientada a objetos, e permite fazer o encapsulamento e a parametrização de formas geométricas, atributos, comportamentos ou alguma combinação destes.

Neste projeto, utilizamos uma PROTO para realizar a intermediação entre o script presente no ambiente virtual, e os dados enviados pela interface ao arquivo de comunicação. Este protótipo fica em um arquivo externo e é importado para o ambiente principal. A princípio, nenhuma alteração é necessária neste arquivo para a sua utilização em diferentes cenas e objetos VRML. Todas os ajustes e configurações podem ser feitas diretamente no script principal, que será apresentado mais adiante.

## **4. Resultados**

---

Todos os testes realizados foram satisfatórios, o objetivo principal foi alcançado. Através da interface JoystickVRML, juntamente com os arquivos intermediários de comunicação, tornou-se possível a interação com objetos presentes no ambiente virtual, com a utilização de comandos enviados por dispositivos externos.

Devido à forma de comunicação de dados, as instruções enviadas pelo dispositivo podem ser utilizadas para apenas um ou vários objetos, bem como, compartilhar comandos de interação, controle e navegação.

Os recursos do DirectInput foram utilizados de forma parcial, somente no que abrange os dispositivos utilizados no desenvolvimento e nos testes de funcionamento, sendo 2 eixos XY, 1 POV de 360° com 9 estados diferentes, além de 12 botões. Todos os comandos podem ser utilizados separadamente, simultaneamente ou em conjunto, neste caso, as combinações de botões.

Esta característica proporciona uma quantidade extremamente grande de instruções de comando e controle. O script do ambiente VRML principal pode identificar, por exemplo, as seguintes situações:

pressionamento do botão 01 – executa a função “abre a porta”    pressionamento do botão 02 – executa a função “fecha a porta”  
pressionamento do botão 03 – “acende a luz”    pressionamento do botão 04 – “apaga a luz”  
pressionamento simultâneo dos botões 01 e 02 – “pula”    pressionamento simultâneo dos botões 01 e 03 – “corre”  
pressionamento simultâneo dos botões 01, 02 e 03 – “chuta” etc...

Com certa destreza, um usuário pode combinar 3, 4, 5 ou mais botões ao mesmo tempo. E se utilizarmos

ainda os 9 estados do POV como botões, temos um total de 21 instruções independentes, que combinadas chegam a um número inesgotável de comandos disponíveis.

## 5. Considerações Finais

---

Através do desenvolvimento deste projeto, e da criação de uma nova interface de comunicação entre dispositivos, scripts e ambientes VRML, abre-se ainda mais o leque de opções para futuras implementações. Inúmeras novas idéias surgiram durante a fase de testes, praticamente eliminando os limites conhecidos, e ampliando os horizontes para uma nova fase de pesquisa e projetos na área de realidade virtual.

Com a possibilidade de envio e recebimento de informações entre mundos virtuais e outras linguagens, novas soluções podem ser encontradas para problemas antigos, bem como, a criação de aplicativos cliente-servidor e sistemas distribuídos.

## Referências Bibliográficas

---

GERSTMANN, Peter. Building 3D virtual environments: an introduction to the creation and implications of single and multiple participant three dimensional virtual environments. Disponível em: . Acesso em: 07 abr. 2006.

GERSTMANN, Peter. Proto Library: A collection to provide assistance in creating VRML worlds. Disponível em: . Acesso em: 10 abr. 2006.

KENNEDY, Chris. Adding USB Joystick Functionality to a Visual Basic Application using Microsoft DirectX. Disponível em: . Acesso em: 06 abr. 2006.

GEOVRML. Oh\_Joy1.0: VRML Joystick Interface. Bind a proto to an object or viewpoint in virtual scene. Disponível em: . Acesso em: 09 abr. 2006.

PARALLELGRAPHICS. EAI - External Authoring Interface support. Disponível em: . Acesso em: 09 abr. 2006.

