



HAL
open science

Usando Modelagem Formal para Especificar Interação em Ambientes Virtuais: Por que?

Luciana Porcher Nedel, Carla Dal Sasso Freitas, Amélie Schyn, David Navarre, Philippe Palanque

► To cite this version:

Luciana Porcher Nedel, Carla Dal Sasso Freitas, Amélie Schyn, David Navarre, Philippe Palanque. Usando Modelagem Formal para Especificar Interação em Ambientes Virtuais: Por que?. Symposium on Virtual and Augmented Reality (SVR 2003), SBC (The Brazilian Computing Society), Oct 2003, Ribeirão Preto, Brazil. pp.81-92. hal-03664741

HAL Id: hal-03664741

<https://hal.science/hal-03664741v1>

Submitted on 11 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Usando Modelagem Formal para Especificar Interação em Ambientes Virtuais: Por que?

Luciana P. Nedel¹, Carla M.D.S. Freitas¹, Amélie Schyn²,
David Navarre², Philippe Palanque²

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

²LIHS – IRIT, Université Toulouse III
118, route de Narbonne – 31062 – Toulouse – France

{nedel, carla}@inf.ufrgs.br, {schyn, navarre, palanque}@irit.fr

Abstract. *The growing number of different devices and interaction techniques provided for virtual reality environments allows many options regarding implementation of applications. Although these new associations between interaction techniques and devices have been tested in experiments with users, they are rarely represented using formal theories. This paper discusses the advantages of using formalism to model interaction in virtual environments. The formalism ICO is presented and used in the modeling of an existing (and tested) application, which has facilitated the discussion of results obtained from the specification.*

Resumo. *A crescente variedade de dispositivos e técnicas de interação para uso em ambientes de realidade virtual gera várias possibilidades de implementação de aplicações. Estas novas associações entre técnicas e dispositivos tem sido normalmente testadas através de experimentos com usuários, mas são raramente especificadas com o auxílio de modelos formais. Este artigo evidencia as vantagens do uso de formalismo na modelagem de interação em ambientes virtuais. O formalismo ICO é aqui apresentado e utilizado na modelagem de uma aplicação já implementada, facilitando a discussão de resultados obtidos com a especificação.*

1. Introdução

Aplicações de realidade virtual apresentam uma série de características que as diferenciam da maior parte das aplicações interativas do tipo WIMP (*Window, Icon, Menu and Pointers*). Enquanto interfaces WIMP são fortemente estáticas, sendo conhecidos, *a priori*, o número de interadores a serem utilizados, seus objetivos e formas de utilização, interfaces desenvolvidas para uso em ambientes virtuais devem considerar outras particularidades, como: continuidade das informações, concorrência, interação a duas mãos e gerência de múltiplos dispositivos, freqüentemente não-convencionais, tanto para entrada de dados como para saída.

A partir destas características, podemos considerar as semelhanças entre sistemas de realidade virtual e sistemas multimodais, que também apresentam uma série de características que dificultam sua modelagem e, conseqüentemente, sua implementação. Como exemplo, pode-se citar o relacionamento entre os diferentes dispositivos de entrada com a lógica do sistema. Em sistemas interativos

convencionais, as formas de entrada são bastante simples (uso do teclado e mouse) gerando eventos também normalmente convencionais (clique, duplo clique, etc.). Em sistemas multimodais, as entradas podem ser múltiplas e complexas (uso de *data gloves*, captadores de posição, movimento da cabeça ou dos olhos, mouse 3D, etc.), sendo que torna-se necessário representar que tipos de eventos serão gerados e como essas entradas serão relacionadas com o resto da aplicação. O tratamento da concorrência também é muito importante, posto que diferentes dispositivos de entrada e/ou saída podem ser (normalmente são) utilizados ao mesmo tempo.

Em linhas gerais, pode-se dizer que a implementação de aplicações de realidade virtual exige a consideração de uma série de questões importantes: o paralelismo entre ações, a espera de uma ação para que outra possa ser executada, a sincronização entre diferentes ações, a fusão de informações provenientes de dispositivos de entrada e a combinação e/ou separação de informações a serem enviadas aos vários dispositivos.

Segundo Smith e Duke (1999), a natureza dos sistemas de realidade virtual contribui para a dificuldade em descrever e modelar interação. Muitos trabalhos encontrados na literatura dedicam-se a introduzir novas técnicas de interação ou descrever configurações particulares de hardware e software assim como seu uso em situações particulares. Vários desses sistemas são avaliados através de estudos experimentais com usuários.

Entretanto, a descrição informal das técnicas de interação dificulta a avaliação de semelhanças entre duas técnicas diferentes, o que acaba muitas vezes por permitir que técnicas básicas de interação sejam muitas vezes “re-inventadas” [Smith e Duke 1999]. Ainda, descrições informais facilitam ambigüidades nas implementações, posto que dão margem a diferentes interpretações por parte dos implementadores. A correta implementação de uma técnica poderia, entretanto, ser obtida através de sucessivas interações entre projetista e implementadores, o que é dificultado nos casos em que os implementadores não mantêm contato com o projetista da técnica.

Este trabalho objetiva evidenciar a utilidade de formalismos na modelagem da interação em ambientes virtuais. Para embasar a argumentação, na Seção 2 é feita uma rápida caracterização da interação em ambientes 3D e a Seção 3 apresenta o formalismo utilizado no estudo de caso relatado na Seção 4. Os resultados aqui apresentados foram gerados a partir da modelagem formal de uma aplicação (xadrez virtual, já implementada [Jacob *et al.* 2002]), o que permitiu que se chegasse às conclusões apresentadas na Seção 5.

2. Interação em ambientes 3D

A interação em ambientes virtuais é realizada com o objetivo de efetuar operações de seleção, manipulação e navegação no ambiente tridimensional (3D). Ações (estímulos motores) realizadas pelo usuário são concretizadas através de técnicas de interação, que correspondem a métodos através dos quais o usuário especifica comandos e dados para o sistema computacional, visando a execução de tarefas. A resposta do sistema se dá através de dispositivos de saída que estimulam os sentidos do usuário (normalmente tato, visão e audição). Tais estímulos são processados e assimilados pelo cérebro resultando o que se denomina de percepção. A percepção e as expectativas criadas com base em experiências anteriores definem como as pessoas interagem com o ambiente. Alguns dos atributos utilizados no processo perceptivo são cor, textura, tamanho, perspectiva, oclusão, sombra, movimento do observador.

No processo de interação humano-computador usando a tecnologia oferecida pelos dispositivos de realidade virtual, busca-se eliminar qualquer estímulo do mundo real através da produção de estímulos virtuais realistas. O objetivo é proporcionar imersão total no ambiente virtual, “desligando” o usuário do mundo real.

Existem várias técnicas de interação para o usuário realizar suas ações no ambiente virtual, sendo algumas apropriadas para determinados tipos de tarefas e outras, não. Técnicas de seleção e manipulação de objetos em ambientes virtuais podem ser classificadas, quanto à metáfora de interação utilizada, como egocêntricas (consideram o usuário como parte do mundo virtual, mantendo a coerência dimensional entre o usuário e os objetos manipulados) e exocêntricas (desconsideram as proporções entre o usuário e os objetos, assumindo que o usuário interage com o ambiente, de fora do seu sistema de referência) [Poupyrev *et al.* 1998].

3. Formalismo ICO

O formalismo ICO (*Interactive Cooperative Objects*) consiste em uma técnica de descrição formal que permite a especificação de sistemas interativos [Bastide *et al.* 1998]. Esta técnica faz uso dos conceitos de orientação a objetos (instanciação dinâmica, classificação, encapsulamento, herança, relacionamento cliente-servidor) para descrever aspectos estruturais ou estáticos de sistemas interativos. Para descrever aspectos comportamentais, utiliza redes de Petri de alto nível [Genrich 1986].

O formalismo gráfico das Redes de Petri é constituído de quatro componentes: lugares (variáveis de estado, representadas por elipses), transições (operadores de troca de estado, representados por retângulos), arcos (ligações entre lugares e transições) e marcas (sinais que indicam o estado atual do sistema). O disparo de uma transição provoca uma mudança no estado do sistema, sendo que uma transição só pode ser disparada se, e somente se, seus lugares de entrada contiverem o número suficiente de marcas requisitado pela transição (definido na modelagem do sistema). Ao ser disparada a transição, as marcas utilizadas são retiradas dos lugares de entrada e novas marcas são depositadas nos lugares de saída da transição.

ICO é dedicado à modelagem e implementação de interfaces baseadas em eventos, utilizando-se de vários objetos que se comunicam para modelar o sistema, onde o comportamento destes objetos e o protocolo de comunicação são descritos através de Redes de Petri. Este formalismo de base, composto de uma técnica de descrição e de um protocolo de comunicação, é denominado CO (Cooperative Objects) e descrito por Bastide *et al.* (2000).

No formalismo ICO, um objeto é representado por uma entidade caracterizada por quatro componentes: um objeto cooperativo com serviços do usuário, uma parte responsável pela apresentação (*rendering*) e duas funções (função de ativação e função de *rendering*) responsáveis pela ligação entre os objetos cooperativos e a parte de apresentação.

Um **objeto cooperativo (CO)** modela o comportamento de um objeto, descrevendo sua reação a estímulos externos, a partir de seu estado interno. Este comportamento, denominado ObCS (Object Control Structure), é descrito através de uma Rede de Petri de alto nível. Um CO oferece dois tipos de serviços a seu ambiente. O primeiro, descrito em CORBA-IDL [OMG 1998], diz respeito aos serviços oferecidos aos outros objetos do ambiente. O segundo, denominado serviço do usuário, fornece uma descrição das ações elementares oferecidas ao usuário.

A **apresentação** de um objeto (*rendering*) representa sua aparência externa. Num contexto convencional, trata-se de um conjunto estruturado de *widgets* organizados em um conjunto de janelas. Um *widget* pode tanto representar uma forma de interagir com o sistema interativo como uma maneira de exibir uma informação relacionada ao sistema interativo.

A interação entre o usuário e o sistema (entrada) ocorre unicamente através desses *widgets* descritos acima. Cada ação do usuário sobre um *widget* pode disparar um serviço de usuário do modelo ICO. A relação entre serviços de usuários e *widgets* é inteiramente descrita pela **função de ativação** que associa a cada dupla (*widget* – ação de usuário) o serviço de usuário disparado. Já a interação entre o sistema e o usuário (saída) visa apresentar ao usuário as trocas de estado do sistema interativo. A **função de *rendering*** mantém a consistência entre o estado interno do sistema e sua aparência externa.

ICO é utilizado para fornecer uma descrição formal do comportamento de uma aplicação interativa, sendo que uma especificação ICO descreve completamente as potenciais interações a que um usuário pode ter acesso em relação a uma aplicação. Esta especificação compreende tanto os aspectos de entrada (como as ações do usuário influenciam o estado interno da aplicação) envolvidos em uma interação como os aspectos de saída (quando e como a aplicação apresenta ao usuário informações pertinentes).

Uma especificação ICO é executável, oferecendo a possibilidade de prototipar e testar uma aplicação antes que ela seja completamente implementada [Navarre *et al.* 2000]. Assim, uma especificação pode ser validada através do uso de ferramentas de análise e de prova desenvolvidas para Redes de Petri e estendidas para levar em conta as especificidades do dialeto de Redes de Petri por ICO.

4. Estudo de Caso: Xadrez Virtual

A aplicação utilizada neste estudo de caso foi originalmente desenvolvida com o objetivo de permitir a experimentação e avaliação de algumas técnicas de seleção e manipulação de objetos 3D em ambientes virtuais [Jacob *et al.* 2002]. Trata-se da representação de um jogo de xadrez (xadrez virtual), onde a tarefa dos jogadores envolve apenas a seleção e movimentação de peças sobre o tabuleiro, sem preocupação com a habilidade cognitiva do jogador, ou com as regras do jogo. A aplicação foi desenvolvida na linguagem C++, utilizando-se a API OpenGL para as funções gráficas de exibição. Nesta seção, sob o ponto de vista da interação, o xadrez virtual será descrito informalmente e, em seguida, utilizando-se do formalismo ICO.

4.1. Descrição informal da aplicação

O xadrez virtual é composto por um tabuleiro com 64 posições (casas) e contém, inicialmente, 32 peças. As peças disponíveis para cada jogador são: o rei, a rainha, 2 bispos, 2 cavalos, 2 torres e 8 peões, sendo que as mesmas são identificadas pelas cores bege e preto. A Figura 1 ilustra as peças e o tabuleiro, em vista superior.

O tabuleiro pode ser visualizado com duas inclinações distintas: vista superior (Figura 1), buscando-se simular uma interface bidimensional (2D) e vista perspectiva (Figura 2), onde procura-se simular o mesmo ângulo de visão experimentado por um jogador em relação a um tabuleiro real. Para passar de uma visualização a outra, o usuário deve se utilizar do teclado, pressionando as teclas 0 (vista 2D) ou 1 (vista perspectiva).

As tarefas a serem realizadas envolvem o deslocamento (horizontal, vertical e diagonal) das peças sem levar em consideração as regras do jogo de xadrez, assumindo apenas que uma peça deve ser colocada sobre uma casa do tabuleiro. O usuário pode selecionar uma peça, deslocá-la e soltá-la sobre uma casa particular. Caso a peça seja colocada no espaço entre duas casas, ela é automaticamente deslocada para a casa mais próxima. Não é permitido ao usuário movimentar as peças para fora do tabuleiro, nem selecionar duas peças simultaneamente. Durante o movimento de uma peça, após a sua seleção, a mesma é suspensa sobre o tabuleiro, simulando o movimento real de pegar uma peça, trocando-a de posição (Figura 2). Como um jogador só pode movimentar uma peça de cada vez, para selecionar uma nova peça, é necessário que todas estejam sobre o tabuleiro.

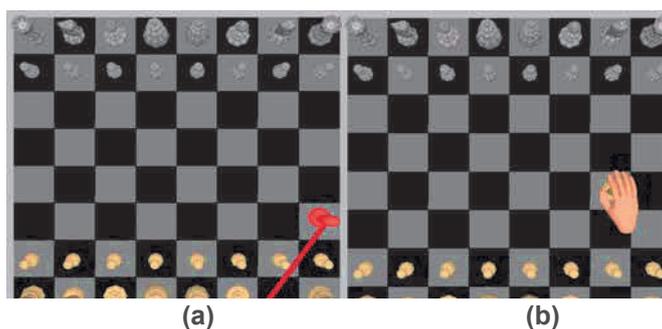


Figura 1. Xadrez virtual: (a) técnica de interação *ray-casting* e (b) “mão virtual”



Figura 2. Vista em perspectiva do xadrez virtual utilizando a técnica de interação da “mão virtual”. Da esquerda para a direita as imagens mostram uma tarefa completa de interação (seleção de uma peça, movimento e colocação).

Dois técnicas de seleção e manipulação são utilizadas (Figura 1): “mão virtual” e *ray-casting*. Na técnica conhecida como “mão virtual”, o usuário é provido com a representação de uma mão no ambiente virtual, sendo que sua posição e orientação são fornecidas ao sistema através de algum dispositivo específico de entrada. Para selecionar um objeto virtual, o usuário deve posicionar a mão “dentro” do mesmo (de forma a interceptá-lo) e informar a aplicação a respeito de sua intenção [Bowman *et al.* 2001]. A técnica *ray-casting* é uma das mais comumente utilizadas para seleção e movimentação de objetos [Poupyrev *et al.* 1998, Bowman e Hodges 1997]. Consiste de um raio infinito e semi-transparente que parte da mão do usuário na direção definida pela orientação da mão virtual. Para selecionar um objeto, o usuário aponta-o e informa a aplicação sua intenção de selecionar, através do dispositivo de entrada disponível.

Em um jogo de xadrez real, a movimentação das peças no tabuleiro é normalmente realizada com a mão, o que sugere a “mão virtual” como forma mais intuitiva de manipulação das peças no xadrez virtual. Como o objetivo deste trabalho não é a comparação entre essas duas técnicas, por razões de espaço será discutida aqui apenas a técnica de interação da “mão virtual”.

Na implementação do xadrez virtual foram utilizados três dispositivos de entrada diferentes: mouse, *data-glove* e captores de movimento. O mouse pode ser considerado um dispositivo de entrada comum, sendo adequado para interações com aplicações baseadas em interfaces gráficas 2D. Muitos consideram que a interação realizada com o mouse é bastante “intuitiva” e “natural” para usuários de computadores com sistemas operacionais baseados em janelas. Um mouse comum foi utilizado nas tarefas de seleção e manipulação das peças do jogo de xadrez, guiando uma mão virtual. Neste caso, o usuário seleciona uma peça posicionando o mouse sobre a mesma e clicando, manipula a peça movimentando o mouse e a recoloca sobre o tabuleiro posicionando o mouse na posição desejada e clicando novamente.

Foi utilizada também uma *data-glove* (luva) modelo 5DT Data Glove 5 (Figura 3). Esta luva possui um sensor de rotação e de inclinação, além de sensores de flexão para os dedos (um sensor de fibra ótica para cada dedo). No xadrez virtual, a luva foi utilizada nas tarefas de seleção e manipulação de objetos, como alternativa ao mouse. A tarefa de seleção de objetos foi implementada através do reconhecimento do gesto da mão sendo fechada, enquanto que a tarefa de largar a peça em uma determinada posição se dá através do gesto de abrir a mão. O movimento da mão fechada com uma peça selecionada permite a movimentação das peças, uma de cada vez.



Figura 3. Data Glove 5DT e captor de movimento Flocks of Birds (Ascension Technology). O captor é preso a *data glove* na altura do punho, ao passo que o emissor de campo magnético encontra-se sobre a mesa, no canto superior direito.

Como a luva não prevê a captura de movimentos de translação, seu uso foi combinado com o de um rastreador de movimento capaz de capturar uma posição no espaço 3D. O equipamento utilizado foi um captor eletromagnético de posição *Flocks of Birds*, comercializado pela Ascension Technology (Figura 3).

4.2. Modelagem formal usando ICO

A partir da descrição informal da interação apresentada na Seção 4.1, concluiu-se que a interface do xadrez virtual necessita gerar os seguintes eventos: *move* (indica a movimentação de uma peça sobre o tabuleiro), *pick* (indica que uma peça foi selecionada), *drop* (notifica que uma peça foi solta) e *idle* (evento gerado quando nenhuma ação estiver sendo executada). Qualquer um destes eventos pode ser produzido por uma variedade de dispositivos de entrada diferentes, neste caso particular pelo mouse ou pela combinação *data-glove* – captor de posição.

A modelagem da aplicação foi estruturada segundo o modelo Arch [Bass *et al.* 1991], ou seja, em cinco níveis: ativação e *rendering* físico, ativação e *rendering* lógico, controle de diálogo, adaptador do núcleo funcional e núcleo funcional. O interesse de uma estruturação deste tipo é facilitar o estudo do impacto de uma troca de dispositivo (nível físico) sobre os níveis seguintes da modelagem. Isto é possível através da modelagem em separado das técnicas de interação e dos dispositivos utilizados (Figuras 4, 5 e 6).

O mecanismo de eventos do ICO prevê que transições possam receber e/ou emitir eventos. O recebimento de eventos é representado por um raio acompanhado de uma identificação (nome) e, opcionalmente, de um conjunto de parâmetros (por exemplo, na Figura 4, a transição *Init* recebe o evento *idle*). Transições sinalizadas como receptoras de eventos, mesmo quando habilitadas só podem ser transpostas quando da ocorrência do evento em questão. Transições ditas emissoras de eventos, por sua vez, são acompanhadas de um raio invertido e igualmente identificado por um nome e um conjunto de parâmetros. Na Figura 4, ao ser disparada a transição *Pick*, além do comportamento normal da Rede de Petri, é emitido o evento *pick(p)*.

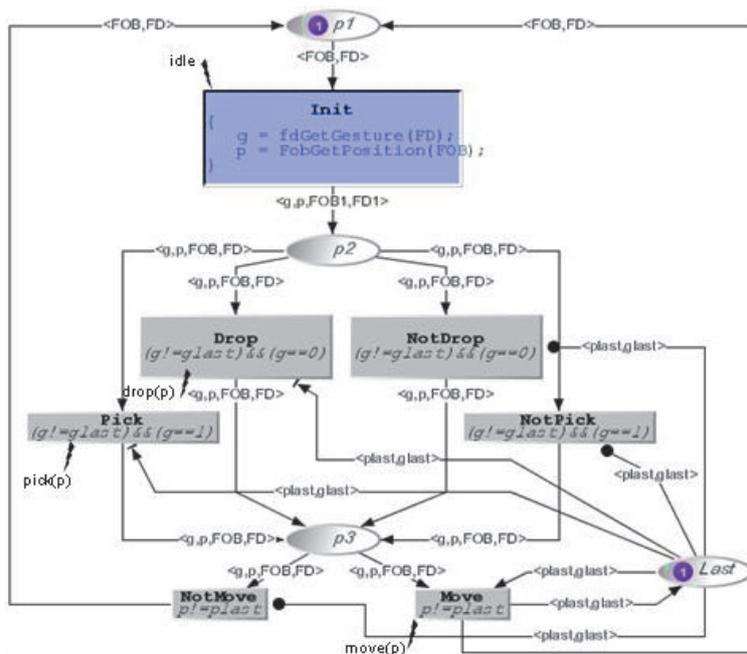


Figura 4. Comportamento de baixo nível com *data glove* e captor de posição

Conforme descrito na Seção 4.1, o xadrez virtual oferece duas diferentes possibilidades de interação com a “mão virtual”: com o mouse, ou utilizando-se uma *data glove* associada a um captor de posição. Estas duas possibilidades de interação são representadas por dois modelos ICO diferentes.

Para produzir os eventos *pick*, *drop* e *move* necessários ao funcionamento da aplicação, o sistema apresentado na Figura 4 deve representar o seguinte comportamento. Cada vez que o evento *idle* (evento produzido automaticamente pela OpenGL sempre que o sistema se encontra em estado de repouso) é produzido, o disparo da transição *Init* permite recuperar, utilizando-se dos sensores associados aos dedos das luvas e do captor de posição, a orientação das falanges dos dedos (permitindo

a identificação de gestos) e a posição da mão no espaço (cálculo das variáveis g e p , respectivamente, na transição).

Inicialmente (transições *Pick*, *NotPick*, *Drop*, *NotDrop*), compara-se a posição corrente da mão com sua posição anterior (este dado é armazenado na marca que está no lugar *Last*). Se esta posição foi alterada e a mão estiver completamente aberta, é produzido um evento *drop(p)*, onde p indica a posição atual da mão. Da mesma forma, se a posição da mão é diferente, a transição *Move* é ativada através da ocorrência de um evento *move(p)*.

Analogamente ao comportamento acima descrito, os mesmos eventos *pick*, *drop* e *move* podem ser produzidos pela utilização do mouse. O funcionamento do mouse, bastante mais simples, é descrito na Figura 5. Sem entrar em detalhes, a cada vez que o mouse é movimentado, um evento do tipo *move* é disparado para ser capturado por este ICO. Qualquer que seja o estado em que se encontra o sistema, o resultado envolve a produção de um novo evento *move*. Em seguida, tanto *pick* como *drop* são eventos associados ao mesmo evento produzido pelo mouse, ou seja, um *click*. A única diferença reside na sequência de cliques produzidos. O primeiro clique gera um *pick*, o segundo, um *drop*, o terceiro, um *pick* e assim sucessivamente.

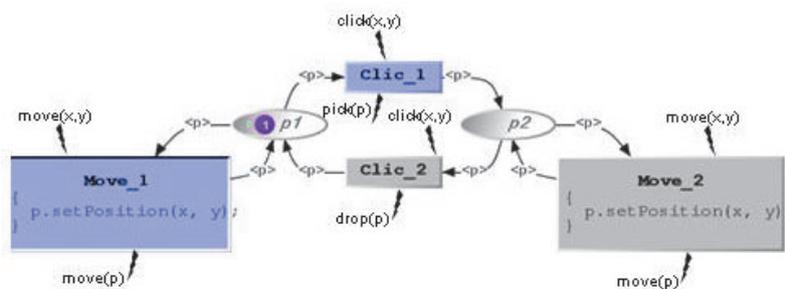


Figura 5. Comportamento de baixo nível com o mouse

No xadrez virtual, o teclado é utilizado apenas para definir (de acordo com a tecla pressionada) se a visualização será feita em 2D (tecla 0) ou em perspectiva (tecla 1). Seu comportamento, definido na Figura 6, é bastante simples. Existem apenas dois estados possíveis para este sistema, correspondendo às duas vistas possíveis e que são definidos em função da última tecla pressionada. A este objeto, diferentemente dos dois outros apresentados anteriormente, observa-se ligação com eventos de *rendering*. Eventos de *rendering* não são diferentes dos utilizados até o presente, a não ser pelo fato de serem utilizados para notificar a um objeto de mais alto nível (ICO de *rendering* lógico) necessidades de alterações no *rendering*. Assim, a cada estado corresponde um evento de *rendering* particular, dado por uma função de *rendering*, conforme apresentado na Tabela 1, que permitirá precisar qual das duas possibilidades de visão será utilizada.

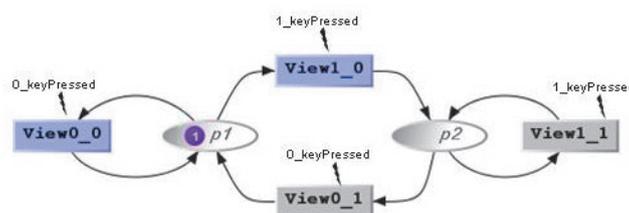


Figura 6. Comportamento do teclado

Tabela 1. Rendering associado ao comportamento do teclado

Lugar	Evento	Evento de <i>rendering</i>
p1	Token_enter	View(0)
p2	Token_enter	View(1)

Conforme mencionado anteriormente, qualquer que seja o dispositivo de entrada escolhido (mouse ou *data glove* – captor de posição), o conjunto de eventos que interessa ao controlador de diálogo permanece os mesmos: *pick(p)*, *drop(p)* e *move(p)*.

Quando um *pick(p)* acontece, a casa *c* do tabuleiro correspondente às coordenadas fornecidas por *p* é recuperada (transição *Pick_1*, na Figura 7). Verifica-se se o lugar *Stock* (contém a totalidade de peças e casas da aplicação) contém uma referência à *c*. Se sim, o usuário pode movimentar a peça correspondente (*Move_2*), soltando-a quando desejar (*Drop_1*). Em caso negativo, espera-se que o usuário recoloca a peça na casa de origem, retornando assim ao estado inicial da interação.

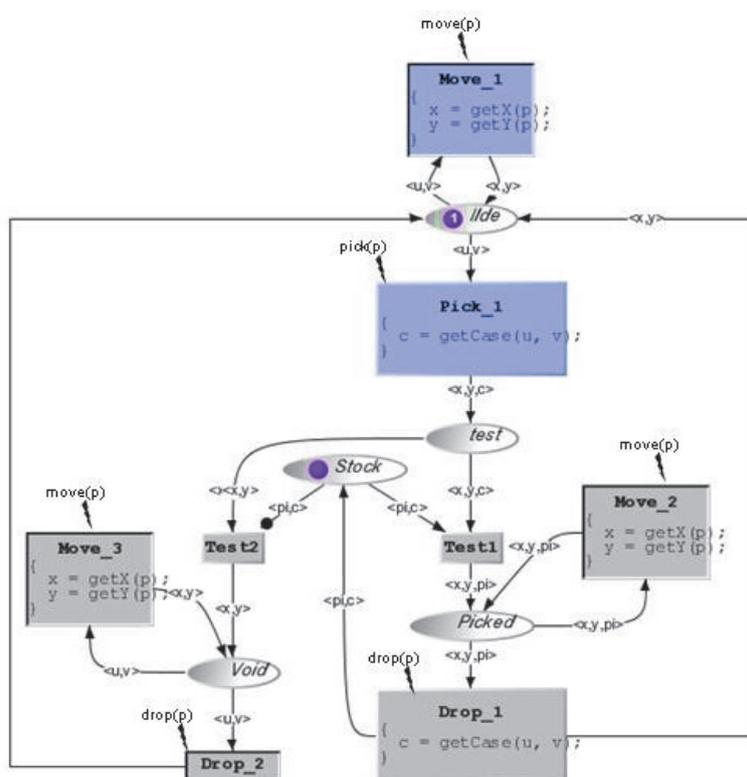


Figura 7. Comportamento do controlador de diálogo

É neste nível que o *rendering* intervém, conforme especificado na Tabela 2. Trata-se de eventos de *rendering* ligados às alterações de estados da rede, recuperados pelos ICO de mais alto nível (ICO de *rendering* lógico e/ou físico), representando os diferentes dispositivos de saída.

Esta delegação de *rendering* a um nível seguinte é necessária porque as informações de *rendering*, neste nível, ainda são incompletas. É possível indicar quais serão as ações a visualizar, mas nada indica que tipo de visualização empregar (2D ou perspectiva), informações estas fornecidas na Tabela 1. Assim sendo, conclui-se que a gerência do *rendering* deverá levar em conta as duas redes apresentadas nas Figuras 6 e 7. Isso se dá através do tratamento dos eventos de *rendering* disparados

em relação às técnicas de interação, como aos diferentes dispositivos de entrada utilizados. Verificou-se que a descrição informal apresentada em trabalho anterior [Jacob *et al.* 2002] era bastante vaga e insuficiente para que a aplicação fosse implementada por alguém que não estivesse envolvido com a sua concepção. Para que a modelagem formal fosse possível, precisou-se recorrer à análise direta do código implementado em C, a fim de relembrar detalhes de concepção. Assim, a modelagem formal incorpora detalhes de interação, evitando que sejam omitidos mesmo que involuntariamente, o que pode ocorrer facilmente na descrição informal. Após ter realizado a modelagem do xadrez virtual com o ICO, sua descrição informal (Seção 4.1) foi re-escrita de forma mais completa.

Durante o desenvolvimento do xadrez virtual, assumia-se que a interação com a aplicação era feita utilizando-se uma técnica tridimensional. No entanto, foi durante a concepção do modelo formal que concluiu-se que a interação, neste caso particular é 2D posto que a altura da mão sobre o tabuleiro não é levada em consideração nunca, sendo sempre constante. Na verdade usou-se uma técnica de interação 2D associada a um cenário 3D, o que só ficou evidente com o formalismo ICO.

Outra observação interessante diz respeito à modelagem da interação em etapas. O formalismo ICO considera separadamente a modelagem da técnica de interação e do dispositivo usado. Assim sendo, consegue-se facilmente isolar os eventos relacionados com a aplicação (no caso do xadrez virtual, apenas *pick*, *move* e *drop*) daqueles gerados pelo dispositivo (e.g., *move* e *click*, no caso do mouse). Assim sendo, o impacto da substituição de um dispositivo de entrada por outro, em uma aplicação, é minimizado, na medida em que modela-se separadamente as entradas e o controle de diálogo. Desta forma, garante-se a separação de todos os aspectos dependentes de dispositivo, o que é importante no desenvolvimento de sistemas de realidade virtual, face à variedade atual de dispositivos.

Agradecimentos

Gostaríamos de agradecer à Liliane J. Jacob e Caroline B. Oliva pela implementação do xadrez virtual. Agradecemos também o suporte financeiro dos projetos CAPES/COFECUB SpiderWeb (projeto 399/02), Competec (SESU-MEC) e CNPq.

Referências Bibliográficas

- Bass, L., Pellegrino, R., Reed, S., Seacord, R., Sheppard, R. e Szezur, M.R. (1991) "The Arch Model: Seeheim Revisited", In: Proceedings of the User Interface Developers' Workshop, version 1.0.
- Bastide, R., Palanque, P., Le Duc, H. e Muñoz, J. (1998) "Integrating Rendering Specification into a Formalism for the Design of Interactive Systems", In: Proceedings of the 5th Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS'98), Springer Verlag.
- Bastide, R., Ousmane, S., Palanque, P. e Navarre, D. (2000) "Formal specification of CORBA services: experience and lessons learned", ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'2000), Minneapolis, Minnesota, USA, ACM Press, pp. 105-117, ACM SIGPLAN Notices, v. 35, n. 10.
- Bowman, D. e Hodges, L. (1997). "An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments". In: Proceedings of Symposium on Interactive 3D Graphics, ACM. pp. 35-38.
- Bowman, D., Kruijff, E., Laviola Jr., J.J. e Poupyrev, I. (2001) "An introduction to 3-D User Interface Design". Presence: Teleoperators and Virtual Environments, vol. 10, no. 1, pp. 96-108.
- Genrich, H.J. (1986) "Predicate/Transition Nets", In: High-Level Petri Nets: Theory and Practice, K. Jensen and G. Rozenberg, Springer Verlag, Berlin, pp. 3-43.

- Lacaze, X., Palanque, P., Navarre, D. e Bastide, R. (2002) "Performance Evaluation as a Tool for Quantitative Assessment of Complexity of Interactive Systems", In: Proceedings of the 9th Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS 2002), Rostock, Dinemark, 2002, Springer Verlag.
- Jacob, L.J., Oliva, C., Nedel, L.P. e Freitas, C.M.D.S. (2002) "Avaliação Experimental de Técnicas de Seleção e Manipulação em Ambientes Virtuais", In: Proceedings of the SBC Symposium on Virtual Reality (SVR 2002), Fortaleza, Brasil.
- Navarre, D., Palanque, P., Bastide, R. e Sy, Ousmane. (2000) "Structuring Interactive Systems Specifications for Executability and Prototypability", In: Proceedings of the 7th Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS 2000), Limerick, Ireland, 2000, Lecture Notes in Computer Science no. 1946, Springer Verlag.
- OMG. (1998) The Common Object Request Broker: Architecture and Specification. CORBA IIOP 2.2/98-02-01, Framingham, MA.
- Poupyrev, I., Weghorst, S., Billingham, M. e Ichikawa, T. (1998) "Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques", Computer Graphics Forum, Eurographics'98 issue, 17(3), pp.41-52.
- Smith, S. e Duke, D. (1999) "The Hybrid World of Virtual Environments", In: Computer Graphics Forum, v. 18, n. 3., The Eurographics Association and Blackwell Publishers.