GeFighters: utilizando gestos para interagir com um jogo de luta

Abstract. *This paper presents GeFighters, a 3D fighting game controlled by gestures. The virtual environment represents a replica of the 70's disco houses, and the fighters are dancers that have their look based on celebrities from that time. GeFighters' major objective is to implement a new type of interaction, utilizing gestures. This is acomplished using an input devices abstraction platform named CIDA. Through this, the application may be controlled by conventional input devices as well as by gesture interaction. The game also takes advantage of control location transparency, so that the gesture capturing process can be distributed over different computers.*

Resumo. *Este artigo apresenta GeFighters, um jogo de luta 3D controlado por gestos. O ambiente virtual é uma réplica das danceterias dos anos 70 e os lutadores são dançarinos que possuem o visual baseado em celebridades da época. O objetivo principal de GeFighters é implementar um novo tipo de interação baseado em gestos, utilizando a plataforma de abstração de dispositivos de entrada CIDA. Através dela, a aplicação pode ser controlada tanto por dispositivos de entrada convencionais quanto pela forma de interação proposta. O jogo também dispõe de transparência de localidade de controle, fazendo com que o processamento da captura dos gestos possa ser distribuído em máquinas diferentes.*

1. Introdução

Na última década o aspecto inovador dos jogos vem diminuindo cada vez mais, tornando as aplicações de entretenimento cada vez mais repetitivas. São lançados sempre os mesmos tipos de jogos, variando apenas na história, parte visual e tipo de interação específica, que geralmente é reaproveitado dos modelos de jogos semelhantes. A interação desses jogos, sejam eles em consoles ou PCs, dá-se através de *joysticks* ou entradas de teclado e *mouse*. Alguns jogos de console inovaram pela captura de gestos através de vídeo, abrindo caminho para esse novo tipo de interação. Todavia, essas aplicações são desenvolvidas para serem jogadas apenas com o dispositivo de captura de gesto específico, o que não permite que o jogador escolha como controle um dispositivo de entrada convencional.

Este trabalho apresenta GeFighters (Gesture Fighters), um jogo de luta 3D que utiliza a plataforma de abstração de dispositivos de interação CIDA (*Chaotic Interaction Devices Abstraction*) [1] para tornar possível a utilização de gestos, assim como dispositivos de entrada comuns, como *mouse* e teclado, de forma transparente para a aplicação.

Ao se usar uma camada de abstração de dispositivos surge a possibilidade de criação de um *joystick* que funciona baseado na captura de gestos, por exemplo, mantendo a interface de interação com a aplicação, que espera um *joystick* como dispositivo de entrada. A plataforma de gerenciamento de dispositivos CIDA foi implementada pelos autores com este propósito. Desta forma, o nível de imersão do

jogador baseia-se no tipo de interação escolhida, de acordo com o dispositivo de entrada utilizado. Por exemplo, se um jogo que precisa de um tempo de resposta rápido para manter uma boa jogabilidade (jogos de luta e corrida) tiver um mapeamento direto entre gestos simples e ações do jogo, a interação via gestos aumentará o nível de imersão do jogador.

A Seção 2 apresenta jogos e aplicações nos quais a interação é realizada por meio de gestos, de forma similar àquela proposta neste artigo, bem como as tecnologias envolvidas no processo de captura e interpretação de gestos. A arquitetura utilizada na construção do GeFighters e o uso da camada de abstração de entrada CIDA são descritos na Seção 3. A Seção 4 explica como foi o processo de criação do personagem do jogo, falando também das ferramentas e métodos envolvidos. A interpretação dos gestos e o mapeamento destes em comandos na plataforma CIDA são explicados na Seção 5. A Seção 6 apresenta as bibliotecas utilizadas no desenvolvimento do jogo e as vantagens da escolha de tais ferramentas. Na Seção 7 são apresentadas as conclusões e idéias de trabalhos futuros que surgiram no decorrer do período de desenvolvimento.

2. Trabalhos Relacionados

A maior parte dos jogos de computador são controlados por dispositivos de entrada convencionais como *joystick*, teclado e *mouse*. Tais jogos não permitem que o jogador faça uso de seus movimentos naturais para interagir com os mesmos, o que implica que ele deve aprender a controlá-los, ou seja, associar seqüências de apertos de botões e movimentos de eixos a ações dentro do jogo. Gestos poderiam fornecer uma forma muito mais intuitiva de interação, uma vez que o usuário já "sabe" como controlar o jogo. Em Decathlete [2], por exemplo, o usuário realmente tem que correr para que seu personagem também o faça. Além disso, usuários com necessidades especiais se beneficiariam por poder controlar os jogos através de piscadas dos olhos ou movimentos da cabeça, já que eles geralmente não possuem a força ou coordenação necessária para utilizar os métodos de entrada convencionais [3].

Por esses e outros motivos, um esforço considerável tem sido empregado em pesquisas na área de reconhecimento de gestos, para utilização na medicina e na indústria, principalmente [4], [5]. Isso leva ao surgimento de produtos comerciais, como o iMatte's iSkia [6] e o Cybernet System's GestureStorm [7]. O primeiro apresenta uma tecnologia que permite que apresentadores interajam com projetores e telas usando gestos, enquanto o segundo permite que apresentadores de previsão do tempo usem o movimento das mãos para ilustrar suas previsões. Diversos teclados virtuais para PDAs (*Personal Digital Assistants*), como o produzido pela Canesta [8].apareceram no mercado como resultado dessas pesquisas. Eles funcionam projetando as teclas numa superfície plana e então capturando os movimentos dos dedos para identificar a tecla "pressionada".

Fabricantes de consoles de videogames também introduziram o conceito de interação com gestos em seus sistemas. A Sega desenvolveu o Activator Ring [9] para o Mega Drive [10]. O anel era constituído por oito seções diferentes e equipado com sensores, os quais correspondiam aos botões de um controle comum desse console. O usuário tinha que pisar no octógono e reproduzir os movimentos pretendidos para o personagem. Mais recentemente, a Sony introduziu a câmera EyeToy [11], que permitia

que alguns jogos fossem controlados utilizando gestos do jogador. O controle do ainda não lançado Nintendo Revolution [12] possui um *tracker* com seis graus de liberdade, trazendo uma forma de interação sem precedentes para os jogos de console.

Apesar das vantagens do uso de gestos no controle de jogos serem bastante claras, existem muitos pontos a serem considerados pelos desenvolvedores. Freeman et al. identificou alguns desafios desse novo tipo de interação, como por exemplo o tempo de resposta (o usuário não deve perceber nenhum atraso entre os seus gestos e a resposta fornecida pelo computador), a confiabilidade dos algoritmos (eles devem ser robustos o suficiente para suportar movimentos imperfeitos e não intencionais) e o custo (dispositivos de interação convencionais apresentam baixo preço). Além do mais, dispositivos comuns de reconhecimento de gestos, como luvas e sensores corporais, são muito intrusivos, e por este motivo torna-se impraticável seu uso diário.

Nesse cenário, o emprego de técnicas e ferramentas tradicionais de Realidade Aumentada (RA) para a captura e reconhecimento de movimentos do usuário é promissor. Bibliotecas de *software* como o ARToolkit utilizam câmeras comuns e de baixo preço, fornecendo algoritmos eficientes de reconhecimento de padrões [13]. Bunchmann et al. recentemente desenvolveu o FingARtips, que visa a interação com objetos virtuais em ambientes de RA baseada no movimento dos dedos da mão [14]. O SymBall [15] é outro exemplo deste tipo de aplicação. Ele foi projetado para ser executado em um celular com câmera e simula um jogo de tênis de mesa onde o usuário move o aparelho para rebater as bolas arremessadas pelo oponente virtual.

3. O Jogo GeFighters

GeFighters é um jogo de luta 3D que teve sua concepção baseada em jogos já **c**onsagrados como Tekken [16] e Dead or Alive [17], ilustrados na Figura 1 a e b, respectivamente. Os principais diferenciais são os personagens envolvidos na luta, que são dançarinos lutadores, o ambiente virtual, que é uma danceteria, e a forma de interação utilizada, através de gestos.

a

b

c

Figura 1. Jogos de luta 3D: a) Tekken b) Dead or Alive c) GeFighters

O jogo apresenta características fortes de comédia e luta. A arena do jogo foi projetada para representar uma pista de dança, com luzes e platéia, assim como ocorre em um ambiente real. A Figura 1 c ilustra o cenário. Os personagens ensaiam passos de dança enquanto não estão sendo controlados pelos jogadores, e quando há interação surgem golpes que visam atingir o oponente. O personagem vencedor é aquele que consegue ganhar dois *rounds* da luta.

GeFighters foi desenvolvido com o objetivo principal de avaliar um novo tipo de interação baseado em gestos. A implementação na aplicação aconteceu através do uso de uma plataforma de gerenciamento de dispositivos de entrada chamada CIDA. Essa plataforma fornece para o jogo transparência na utilização de dispositivos de entrada, assim como no tipo de interação utilizado. Mais detalhes dessa interação podem ser encontrados na Seção 5.

O uso de CIDA permite que todo o sistema do jogo seja distribuído em até três máquinas: uma com o processamento do jogo em si e duas funcionando como os controles do mesmo (capturando e interpretando os movimentos). Portanto, além da flexibilidade de dispositivo, essa plataforma ainda permite a transparência de localidade da entrada para a aplicação.

A arquitetura do GeFighers encontra-se dividida em dois grandes módulos, como ilustrado na Figura 2. O módulo do OGRE (*Object-oriented Graphics Rendering Engine*) [18] é responsável pela parte visual do jogo, renderizando todos os componentes através do DirectX ou OpenGL. O módulo de CIDA se encarrega da interação com os jogadores, e esses controlam os personagens por meio de gestos. Um dos *plugins* da camada de abstração de dispositivos CIDA mapeia a entrada no *joystick* virtual e um *plugin* de rede faz a conexão entre os usuários de diferentes máquinas. Detalhes relacionados com a interação do usuário com o jogo serão descritos na Seção 5.

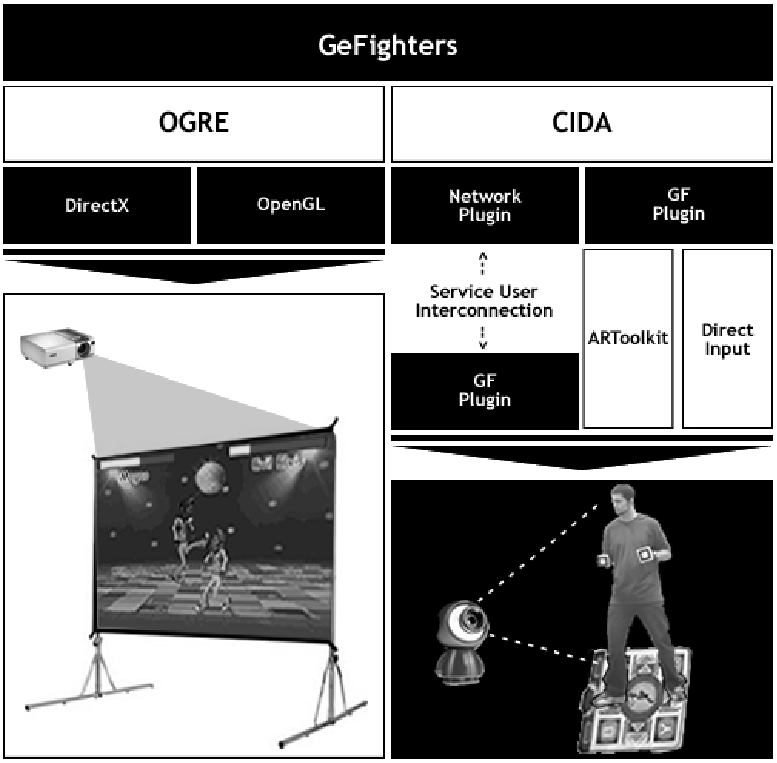

Figura 2. Arquitetura do GeFighters

4. Modelagem do Personagem

Antes do processo de modelagem do personagem, foi realizada uma pesquisa com o intuito de coletar referências dos figurinos utilizados pelas pessoas nas décadas de 60 e 70. Esta pesquisa foi focada principalmente nas roupas, acessórios e penteados comuns àquela época. Os dados obtidos nessa coleta serviram como base para a modelagem e texturização do personagem, tornando assim o trabalho mais fiel à realidade.

A ferramenta 3D Studio Max [19] foi utilizada na construção do modelo do personagem. Para a modelagem do dançarino e criação das texturas foram utilizadas, como referência, fotos de pessoas e roupas. O processo de animação foi um pouco mais complicado que os de modelagem e texturização, pois ele necessita que previamente seja criado um esqueleto (estrutura hierárquica de *bones*), conforme mostrado na Figura 3. O esqueleto foi construído baseando-se nas articulações dos ossos humanos, o que possibilitou que as animações pudessem ser similares aos movimentos reais. Com o esqueleto totalmente construído, o próximo passo foi ligar a pele aos ossos (esta técnica é chamada de *Skinning*), de modo que ela seguisse os movimentos dos ossos durante a animação. Após todas essas etapas os modelos foram exportados e posteriormente carregados pelo jogo.

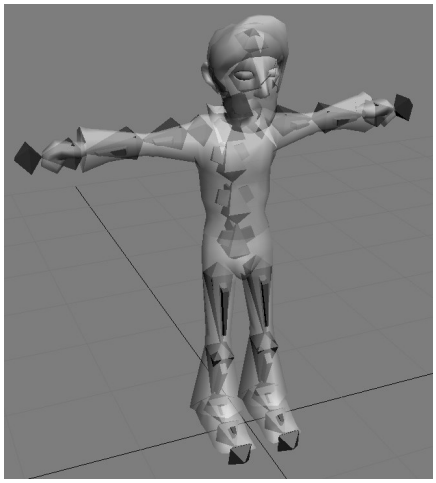

Figura 3. *Bones* do esqueleto cobertos pela pele do personagem

O personagem Magao teve sua concepção inspirada em dançarinos e cantores. Ele foi concebido baseando-se em artistas famosos como John Travolta (no filme Grease), Elvis Presley (com suas roupas características) e Michael Jackson. Jaqueta de couro, calça boca de sino e cabelo com topete estão entre alguns itens herdados desses artistas pelo personagem. Além da aparência, também foram herdados movimentos característicos (passos de dança) desses artistas, como por exemplo, a "andada" para trás de Michael Jackson (*Moonwalking*).

A Figura 4 ilustra os movimentos e ações do personagem: parado, andando para frente, andando para trás, agachando, atacando com soco, atacando com chute, pulando diretamente para cima, dando cambalhota no ar e atacando com soco e chute aéreos, respectivamente.

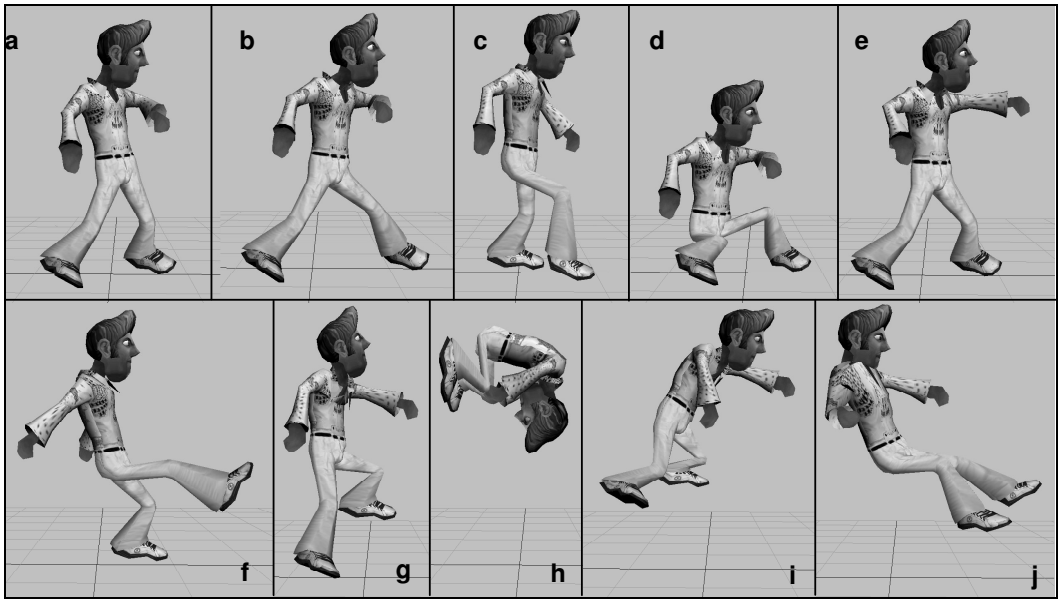

Figura 4. Movimentos do personagem Magao

5. Interação com Gestos

À medida que os computadores pessoais e consoles de jogos vêm aumentando significativamente seu poder de processamento, novos estilos de interação podem ser considerados. Parte destas novas formas de interação surgem de pesquisas na área de realidade virtual (RV), utilizando dispositivos como luvas, capacetes, *trackers*, entre outros.

A imersão tem sido levada em consideração por aplicações que pedem formas de interação não convencionais. Nestes ambientes a interação deve ser feita da forma mais natural possível para conservar seu aspecto imersivo.

Uma das técnicas utilizadas para interagir com ambientes imersivos é a interação com gestos. Esta pode ser feita utilizando luvas, *trackers* e até mesmo dispositivos hápticos. Outra maneira de se identificar gestos é através da captura de imagens, utilizando uma câmera. O vídeo é capturado e processado quadro a quadro para extrair padrões relativos às características desejadas e, a partir de uma seqüência de padrões reconhecidos, tem-se um conjunto de dados que podem ser interpretados como um gesto. Esta técnica é utilizada em alguns jogos como "EyeToy: Kinetic" [20], um jogo onde um *personal trainer* virtual sugere uma série de exercícios e verifica se o jogador está se exercitando corretamente. Este jogo foi projetado para o console PlayStation 2 e utiliza o periférico "EyeToy USB Camera" para capturar o vídeo que identificará as deficiências do jogador e será base do julgamento do treinador virtual. Este periférico foi pouco difundido, porém o estilo de interação oferecido pelos jogos que o utilizam vem crescendo bastante em conjunto com aplicações de RA e Realidade Misturada (RM). Nessas aplicações, o objetivo principal é reunir objetos virtuais em um mundo real, utilizando padrões de marcadores como forma de interação.

Algumas ferramentas e bibliotecas foram desenvolvidas na área de RA, RM e visão computacional, entre elas o ARToolkit, MXRToolkit [21] e OpenCV [22], respectivamente. Todas estas podem ser utilizadas para detectar padrões, sendo a última mais genérica, onde um processamento específico pode ser realizado para identificar padrões não convencionais.

Padrões convencionais são aqueles que seguem o estilo dos marcadores, compostos geralmente por quadrados com bordas pretas e símbolos monocromáticos em sua região central. Esses símbolos podem compor figuras indexadas por uma ferramenta de geração de padrões ou fazer parte de um conjunto de marcadores baseados em "ID" (*ID Based Markers*) [23]. Ao contrário desse tipo de padrão, os não convencionais podem ser formados por qualquer informação contida no ambiente, como por exemplo, faces, dedos, luminosidade, simetrias, contornos, entre outros.

No GeFighters também é utilizado um tipo de interação com gestos através do ARToolkit (mais detalhado na Seção 6.2), uma biblioteca de reconhecimento de marcadores. Dois deles são responsáveis por dar origem aos movimentos do personagem, como ilustra a Figura 5. O usuário deve segurar um marcador em cada mão, de forma que os padrões sempre estejam voltados para a câmera.

Figura 5. Direção dos eixos X e Y de acordo com o posicionamento dos marcadores

Para facilitar a explicação de como o mapeamento é realizado, o padrão da mão direita foi denominado G e o da mão esquerda F. Esses dois padrões são mapeados em dois eixos de um *joystick*, o que pode ser implementado de várias maneiras. Optou-se por usar o vetor formado pela posição relativa entre os dois marcadores, como um manche de *joystick*. Dessa forma, tem-se informação sobre ambos os eixos X e Y. A plataforma CIDA (mais detalhes na Seção 6.3) é responsável por mapear esse vetor de posição relativa nas informações dos eixos, fazendo com que a aplicação acesse o dispositivo de entrada como um *joystick* virtual. A biblioteca de detecção de padrões fornece informações sobre a localização espacial e a rotação dos padrões, porém optou-se por usar apenas duas dimensões, uma vez que o mapeamento ocorre sobre apenas dois eixos do *joystick*.

Todos os movimentos do personagem são baseados na posição relativa entre os marcadores, conforme citado anteriormente. O marcador F funciona como ponto de referência (ou seja, funciona como se estivesse parado) e a posição do marcador G define o movimento a ser interpretado. Por exemplo, caso o usuário esteja segurando o marcador G mais à frente do marcador F, significa que o eixo X possui o valor 1 (considerando que o mesmo varia entre -1 e 1, da esquerda para a direita). Caso o marcador G esteja posicionado abaixo do marcador F, significa que o eixo Y apresenta o valor -1.

Pode-se ter uma idéia melhor do posicionamento e mapeamento dos marcadores através da Figura 5. Ela ilustra todas as posições que podem ser mapeadas nos dois eixos. As imagens que se encontram nas diagonais representam as posições compostas, formadas quando ambos os valores dos eixos X e Y são diferentes de 0. A imagem central (quando os dois marcadores estão lado a lado, na mesma altura) indica o estado de repouso do controle, ou seja, o personagem não está em movimento no ambiente

virtual. Caso o jogador queira realizar um movimento de "pular para trás" com o personagem, ele deverá posicionar os marcadores conforme a primeira imagem do canto superior esquerdo, no qual o padrão G está localizado atrás e acima do padrão F. Caso o jogador queira que o personagem se abaixe, deverá posicionar o marcador G abaixo do marcador F, conforme pode ser visto na imagem central da parte inferior.

O mapeamento dos eixos foi implementado de forma que existe uma região intermediária entre os estados de movimento, que é denominada de vazio (*deadzone*). Essa região é a responsável por fazer com que não haja oscilações durante a transição entre dois estados. Isso é evitado pelo fato do movimento não ser alterado na região de vazio. Dessa forma, para o usuário passar do estado "parado" para o estado "direita", deverá passar pela zona de transição vazia. Caso os valores de localização do marcador fornecidos pela biblioteca oscilem, devido a problemas com a captura da imagem, o estado atual do movimento é mantido. Em resumo, os estados só variam de um para outro através de uma movimentação significativa do marcador.

Além dos eixos, um tapete de dança foi utilizado para o mapeamento dos botões do *joystick*. O usuário deve pisar nos sensores do tapete para controlar as ações de ataque do personagem. Através dos botões pode-se socar, chutar e até executar golpes especiais, dependendo da seqüência em que eles são pressionados.

O tapete de dança é composto por 6 botões e dois eixos, dos quais os eixos não são utilizados, uma vez que a informação referente aos valores de X e Y é obtida com os marcadores. Ele foi originalmente desenvolvido para uso com o console Playstation, mas também pode ser conectado ao computador através da porta paralela.

6. Bibliotecas Utilizadas

As bibliotecas OGRE, ARToolkit e CIDA foram utilizadas no desenvolvimento do GeFighters. Elas foram escolhidas, dentre outros motivos, por tornar a aplicação melhor estruturada e diminuir seu tempo de desenvolvimento.

6.1. OGRE

O jogo utiliza como motor de renderização a biblioteca OGRE. Ele é um motor gráfico *open-source* que funciona na maioria das plataformas existentes e oferece uma vasta gama de *plugins*, ferramentas e *add-ons* que favorecem a criação de diversos tipos de aplicações gráficas, empregando vários conceitos inerentes ao desenvolvimento das mesmas.

O propósito do OGRE não é ser um motor de jogos; ele é um motor de renderização genérico que pode ser incorporado a bibliotecas de tratamento de entradas, processamento de som e plataformas que disponibilizem algoritmos de inteligência artificial, compondo assim um *kit* de desenvolvimento completo para jogos 3D.

O uso dessa biblioteca justifica-se pelo fato da mesma possuir uma boa documentação, já estar bastante difundida pela comunidade e prover uma interface de programação de alto nível que diminui o tempo de desenvolvimento da aplicação.

6.2. ARToolkit

O ARToolkit é uma biblioteca criada para dar suporte ao reconhecimento de padrões em imagens. Ele é utilizado amplamente em aplicações de RA por possuir um componente para criação de marcadores e outro para captura de vídeo. Além dessas funcionalidades, também devem ser ressaltados o reconhecimento de padrões, extração de suas características (por exemplo, orientação e posição em relação ao ambiente) e sobreposição de objetos virtuais em imagens adquiridas do mundo real.

Essa biblioteca é utilizada pelo módulo de detecção de gestos do GeFighters, de forma a reconhecer os marcadores, que serão rastreados para determinar o movimento dos personagens. Ela foi escolhida por apresentar as vantagens citadas anteriormente.

6.3. CIDA

CIDA é uma plataforma de gerenciamento de dispositivos de entrada cuja principal característica é oferecer uma camada de abstração entre sistemas interativos e esses dispositivos. A plataforma disponibiliza para a aplicação, em tempo de execução, todos os dispositivos que possuem as características necessárias para interagir com a mesma, como botões, eixos, graus de liberdade, entre outros. Os dispositivos em CIDA são agrupados em quatro classes distintas, de acordo com as suas características funcionais: *Keyboard*, *Joystick*, *Tracker* e *Pointer Device*.

O suporte a dispositivos de entrada é viabilizado através de *plugins* específicos, que determinam quais são as características de cada classe que o dispositivo oferece. Esta característica simplifica a adição de novos dispositivos de interação a sistemas baseados na plataforma, pois o código-fonte da aplicação não precisa ser modificado. Além disso, um mesmo dispositivo pode ser enxergado de formas diferentes pela aplicação, uma vez que as suas capacidades estão especificadas em cada *plugin*. Por exemplo, um teclado convencional poderia ser visto pela aplicação como um dispositivo da classe *Keyboard* ou como dois dispositivos da classe *Joystick*, dependendo do *plugin* utilizado. A Figura 6 ilustra este exemplo.

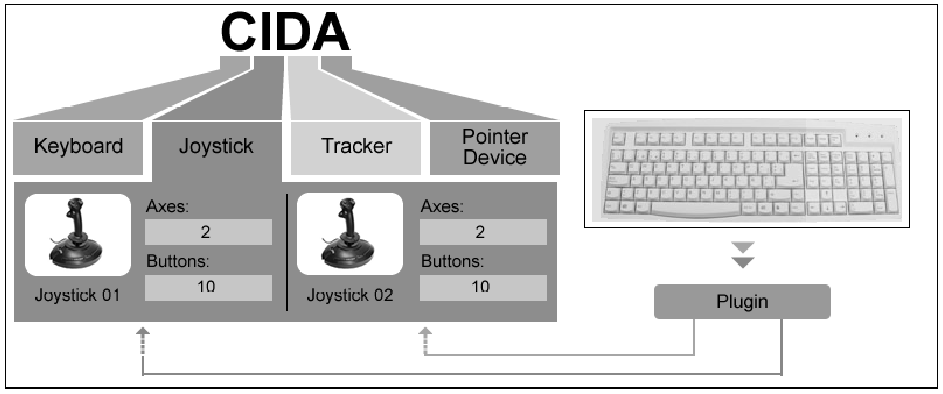

Figura 6. Teclado reinterpretado como dois *joysticks*

A plataforma também oferece transparência de conectividade, pois os dispositivos podem estar conectados a um computador diferente daquele que executa a aplicação, sem que o código-fonte precise ser modificado. Essa característica facilita consideravelmente o gerenciamento de dispositivos de entrada em ambientes de RV complexos, nos quais o uso de vários computadores é necessário para garantir o

processamento em tempo real dos dados de entrada. Além disso, o desenvolvedor não precisa se preocupar com o uso de *device drivers* específicos para determinados periféricos, pois a plataforma oferece à aplicação uma visão uniforme das capacidades de cada dispositivo.

A plataforma CIDA foi utilizada no desenvolvimento do GeFighters para permitir que dois jogadores pudessem interagir remotamente, além de tornar possível o uso de controladores de jogo tradicionais como substitutos do controle através de marcadores do ARToolkit.

7. Conclusões e Trabalhos Futuros

Este artigo apresentou GeFighters, um jogo de luta projetado para testar um ambiente virtual com interação através de gestos. Nele foi descrito o funcionamento do jogo, bem como o processo de modelagem, ambientação e animação dos personagens. O principal foco do artigo foi estudar formas de interação com gestos e propor um novo tipo de interação, baseado em marcadores. A plataforma de abstração de dispositivos de entrada CIDA mapeia a informação adquirida pela imagem captada em entradas de um *joystick* virtual. CIDA permite que a qualquer momento a aplicação deixe de ser comandada por gestos e passe a ser comandada por um *joystick* comum. Também foram apresentadas as ferramentas e bibliotecas utilizadas na implementação do jogo e sua ligação com a captura de gestos e a reprodução do resultado dos comandos.

Como trabalhos futuros, estão previstos o mapeamento de gestos independentes da localização relativa dos marcadores e a adição de novos personagens ao jogo. Poderão ser definidos novos tipos de gestos, assim como interpretações diferentes daquelas já existentes. Também está prevista uma comparação entre alguns tipos de interação via gestos com relação ao desempenho, tempo de resposta do algoritmo de reconhecimento e conveniência/facilidade para o usuário.

8. Referências

[1] Suprimido para a revisão cega.

[2] W.T. Freeman, D.B. Anderson, P.A.Beardsley, C.N. Dodge, M.W.C.D. Roth, e W.S Yerazunis, "Computer vision for interactive computer graphic", *IEEE Computer Graphics and Applications*, 18(3), IEEE Press, New York, USA, 1998, pp. 42-53.

[3] C.E. Steriadis, e P. Constantinou, "Designing Human-Computer Interfaces for Quadriplegic People", *ACM Transactions on Computer-Human Interaction*, 10(2), ACM Press, New York, USA, 2003, pp. 87-118.

[4] K. Köchy, M. Krauss, e P. Neumann, "Interactive Manipulation of Realtime Visualisation from Medical Volume Data by using 2-Handed VR-Techniques", *EuroPACS*, 1998, pp. 221-224.

[5] B.A. Myers, "A Brief History of Human Computer Interaction Technology", *ACM Interactions*, 5(2), ACM Press, New York, USA, 1998, pp. 44-54.

[6] iSkia Projector. Disponível em: Imatte site. URL: http://www.imatte.com, visitado em Janeiro 2006.

[7] GestureStorm Weather Map Management System. Disponível em: Cybernet Systems Corporation site. URL: http://www.cybernet.com, visitado em Janeiro 2006.

[8] Canesta's Virtual Keyboard for PDAs. Disponível em: Canesta Inc. site**.** URL: http://www.canesta.com, visitado em Janeiro 2006.

[9] Activator Ring. Disponível em: Sega site. URL: http://www.sega.co.jp, visitado em Janeiro 2006.

[10] Mega Drive. Disponível em: Sega Mega Drive/Sega Genesis Wikipedia site. URL: http://en.wikipedia.org/wiki/Sega_Genesis, visitado em Janeiro 2006.

[11] EyeToy USB Camera. Disponível em: Sony Playstation site. URL: http://www.us.playstation.com, visitado em Janeiro 2006.

[12] Nintendo Revolution. Disponível em: Nintendo site. URL: http://www.nintendo.com/home, visitado em Janeiro 2006.

[13] ARToolkit. Disponível em: Human Interface Technology Lab site. URL: http://www.hitl.washington.edu/artoolkit, visitado em Janeiro 2006.

[14] V. Buchmann, S. Violich, M. Billinghurst, e A. Cockburn, "FingARtips – Gesture Based Direct Manipulation in Augmented Reality", *International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, 2004, pp. 212-221.

[15] M. Hakkarainen, e C. Woodward, "SymBall - Camera Driven Table Tennis for Mobile Phones", *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, 2005, pp. 1-2.

[16] Tekken. Disponível em: Tekken Official site. URL: http://www.tekken-official.jp/, visitado em Janeiro 2006.

[17] Dead or Alive. Disponível em: Dead or Alive site. URL: http://www.deadoralivegame.com/, visitado em Janeiro 2006.

[18] OGRE. Disponível em: OGRE site. URL: http://www.ogre3d.org, visitado em Janeiro 2006.

[19] 3D Studio Max. Disponível em: Autodesk site. URL: http://www.autodesk.com/3dsmax, visitado em Janeiro 2006.

[20] EyeToy: Kinetic. Disponível em: EyeToy: Kinetic site. URL: http://www.eyetoykinetic.com, visitado em Janeiro 2006.

[21] W. Bath, and J. Paxman, "UAV Localisation & Control Through Computer Vision", *Australian Robotics & Automation Association*, 2005.

[22] Open Source Computer Vision Library. Available: Intel Corporation site. URL: http://www.intel.com/technology/computing/opencv/, visitado em Janeiro 2006.

[23] M. Fiala, "ARTag, an Improvement Marker System Based on ARToolkit", *NRC Technical Report*, National Research Council of Canada, Canada, 2004, pp. 1-37.