DANIEL CALIFE

ROBOT ARENA: UMA INFRA-ESTRUTURA PARA O
DESENVOLVIMENTO DE JOGOS COM REALIDADE AUMENTADA
ESPACIAL

SÃO PAULO
2008

DANIEL CALIFE

ROBOT ARENA: UMA INFRA-ESTRUTURA PARA O
DESENVOLVIMENTO DE JOGOS COM REALIDADE AUMENTADA
ESPACIAL

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do título de
Mestre em Engenharia.

SÃO PAULO
2008

DANIEL CALIFE

ROBOT ARENA: UMA INFRA-ESTRUTURA PARA O
DESENVOLVIMENTO DE JOGOS COM REALIDADE AUMENTADA
ESPACIAL

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para a obtenção do título de
Mestre em Engenharia.

Área de Concentração:
Engenharia Elétrica – Departamento de
Computação e Sistemas Digitais.

Orientador:
Prof. Livre-Docente
Romero Tori

SÃO PAULO

2008

FICHA CATALOGRÁFICA

Calife, Daniel

**Robot Arena : uma infra-estrutura para o desenvolvimento de jogos com realidade aumentada espacial / D. Calife. -- São Paulo, 2008.
110 p.**

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Jogos eletrônicos 2. Computação gráfica 3. Interface homem-computador 4. Realidade virtual I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II. t.

para Gi.

AGRADECIMENTOS

Ao meu orientador, professor Romero, por toda a sua ajuda ao longo deste trabalho.

Ao Alexandre e ao Diego por toda a ajuda e trabalho árduo desde o começo deste projeto.

Ao Fernando e ao Marco pela ajuda no desenvolvimento dos protótipos dos jogos TanSpace e FootBot ARena.

Ao João, Ricardo e Roberto por toda a ajuda técnica, dicas e revisão deste trabalho.

À Mariza pela presteza e eficiência para resolver as dificuldades administrativas do Mestrado.

A todos aqui citados, aos alunos, professores e integrantes do Interlab pela sua amizade ao longo destes anos.

Ao CNPq pela bolsa de pesquisa.

Ao ItauCultural pelo prêmio Rumos Arte Cibernética 2006, na carteira de apoio à pesquisa acadêmica.

À Gi e a minha mãe, Roseli, por todo seu apoio, amor e compreensão.

“E assim escondi-me atrás da porta, para que a Realidade, quando entra, me não veja. Escondi-me debaixo da mesa, donde, subitamente, prego sustos à Possibilidade. De modo que desligo de mim, como aos dois braços de um amplexo, os dois grandes tédios que me apertam – o tédio de poder viver só o Real, e o tédio de poder conceber só o Possível.

Triunfo assim de toda a realidade.”

Fernando Pessoa, Livro do Desassossego.

RESUMO

Atualmente Jogos Eletrônicos possuem uma grande relevância no setor econômico, na computação e na pesquisa acadêmica, não se limitando apenas a aplicações de entretenimento. Na busca por inovação na jogabilidade e nas interfaces interativas dos jogos eletrônicos, a Realidade Aumentada é uma das novas fronteiras a serem exploradas. Este trabalho apresenta a pesquisa, desenvolvimento e teste de uma infra-estrutura, denominada Robot ARena, para o desenvolvimento de jogos inovadores utilizando a Realidade Aumentada e as novas interfaces que esta tecnologia possibilita. Baseada em interfaces horizontais e composta por elementos de *hardware* e *software*, a Robot ARena possui uma arquitetura flexível que possibilita diversas formas de interação e de visualização. Outro aspecto previsto neste projeto é o controle e a comunicação de robôs, que podem ser utilizados como avatares nos jogos. Os conceitos envolvidos, Jogos Eletrônicos e Realidade Aumentada Espacial, são discutidos, bem como os trabalhos que inspiraram alguns dos aspectos desta infra-estrutura. Um protótipo da Robot ARena foi implementado e testado e utilizado no desenvolvimento de dois protótipos de jogos, FootBot ARena e TanSpace, que trazem novos desafios a esta infra-estrutura, como explorar a influência de objetos virtuais aos objetos reais e a aplicação de interfaces tangíveis.

Palavras Chave: Jogos Eletrônicos, Computação Gráfica, Interface homem-computador, Realidade Virtual.

ABSTRACT

Nowadays Electronic Games have a great importance in the economic sector, in computing and in academic research, not limited only to entertainment applications. In the search for innovation in gameplay and in the interactive interfaces of electronic games, Augmented Reality is one of the new frontiers to be explored. This work presents the research, development and testing of an infrastructure, called Robot ARena, for the development of innovative games using the Spatial Augmented Reality and new interfaces that this technology allows. Based on horizontal interfaces with elements of hardware and software, Robot ARena has a flexible architecture that enables different ways of interaction and visualization. Another aspect planned for this project is the control and communication of robots, which can be used as avatars in games. The concepts involved, Electronic Games and Spatial Augmented Reality, are discussed, as well as the work that inspired some aspects of this infrastructure. A Robot ARena prototype was implemented, tested and used in the development of two games prototypes, FootBot ARena and TanSpace, which brings new challenges to the infrastructure, such as exploration of the influence of virtual objects in real objects and the application of tangible interfaces.

Keywords: Electronic Games, Computer Graphics, Computer-Human Interface, Virtual Reality.

LISTA DE ILUSTRAÇÕES

Figura 1 - Realidade Aumentada, robô real interagindo com baterias e barril virtuais.	21
Figura 2 - O contínuo Realidade – Virtualidade, adaptada de (MILGRAM et al., 1994).	22
Figura 3 - Classificação dos Dispositivos de Exibição, adaptada de (BIMBER; RASKAR, 2005).	24
Figura 4 - Exemplo de dispositivo “ <i>Window in the World</i> ” aplicado no projeto Kobito (AOKI et al., 2005).	25
Figura 5 - Exemplo de aplicação do dispositivo <i>Optical See-through</i> Espacial, extraída de (LASER, 2007).	26
Figura 6 - Exemplo de aplicação um Dispositivo Espacial Baseado em Projeção, extraída de (LOW et al., 2001).	27
Figura 7 - Relação entre componentes envolvidos na projeção geométrica, adaptada de (BIMBER, RASKAR, 2005).	29
Figura 8 - Estratégia de renderização para projeção geométrica, adaptada de (BIMBER; RASKAR, 2005).	30
Figura 9 – <i>Game engine</i> na estrutura de um jogo, adaptada de (LEWIS; JACOBSOM, 2002).	41
Figura 10 – Principais componentes de um <i>Game Engine</i>	42
Figura 11. O jogo <i>indoor</i> SENAC ARHockey, extraída de (VIEIRA et al., 2006).	48
Figura 12 - As três camadas da arquitetura do enJine, extraída de (BERNARDES JR; NAKAMURA; TORI, 2007).	50
Figura 13 - Os principais pacotes do enJine, extraída de (BERNARDES JR; NAKAMURA; TORI, 2007).	51
Figura 14 – Arquitetura da camada de entrada do enJine, adaptado de (BERNARDES JR; NAKAMURA; TORI, 2007).	52
Figura 15 - Arquitetura da integração enJine + jARToolKit, adaptada de (TSUDA et al., 2007).	53
Figura 16 - O dispositivo PIP em uso (STUDIERSTUBE, 2007).	54
Figura 17 – A arquitetura do Studierstube, adaptada de (STUDIERSTUBE, 2007). ..	55

Figura 18 - Arquitetura e conceito da aplicação Kobito, adaptada de (AOKI et al., 2005).	57
Figura 19 – Robôs em um ambiente virtual de <i>Augmented Coliseum</i> , extraída de(SUGIMOTO et al., 2005).	58
Figura 20 - Estrutura de um jogo com a camada da Robot ARena, baseada em (LEWIS; JACOBSOM, 2002).	61
Figura 21 – Arquitetura da Robot ARena.	62
Figura 22 – A primeira configuração da Robot ARena.	67
Figura 23 – Versão final da Robot ARena.	68
Figura 24 – Primeira versão do robô.	69
Figura 25. Módulo de comunicação da primeira versão do robô.	69
Figura 26 – Versão final do robô, montado com o Lego NXT.	70
Figura 27 – Marcador colorido, montado com blocos Lego.	70
Figura 28 – Dispositivo de rastreamento infravermelho.	71
Figura 29 – Marcador Lego amarelo.	71
Figura 30 – Diagrama de classes do subsistema de Visualização.	72
Figura 31 – Diagrama de classes do subsistema de Rastreamento e Registro.	74
Figura 32 – (a) imagem capturada pela câmera; (b) imagem após correções.	75
Figura 33 – Diagrama de classes do subsistema de Elementos Reais.	76
Figura 34 – Diagrama de classes do subsistema de Comunicação e Controle do Robô.	78
Figura 35 – Imagem da prova de conceito da Robot ARena.	79
Figura 36 – Protótipo FootBot ARena em execução.	83
Figura 37 – Elementos do jogo TanSpace.	84
Figura 38 – Protótipo TanSpace em execução.	85

LISTA DE TABELAS

Tabela 1 – Características dos Jogos Tradicionais e Jogos Eletrônicos.....	47
Tabela 2 – Relação de Características por Projeto.	59

LISTA DE ABREVIATURAS E SIGLAS

ANSI	<i>American National Standards Institute</i>
API	<i>Application Programming Interface</i>
AR	<i>Augmented Reality</i>
cm	Centímetro
CMOS	<i>Complementary Metal-Oxide Semiconductor</i>
CMYK	<i>Cyan, Magenta, Yellow, Black</i>
CRT	<i>Cathode Ray Tube</i>
DBC	<i>Display-Based Computing</i>
DLL	<i>Dynamic Link Library</i>
DS	<i>Dual Screen</i>
DWARF	<i>Distributed Wearable Augmented Reality Framework</i>
GB	<i>Giga Byte</i>
GHz	<i>Giga Hertz</i>
GPL	<i>General Public License</i>
HMD	<i>Head Mounted Display</i>
HW	<i>Hardware</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
INTERLAB	Laboratório de Tecnologias Interativas
IR	<i>Infrared</i>
JNI	<i>Java Native Interface</i>
JSR	<i>Java Specification Request</i>
LCD	<i>Liquid Crystal Display</i>
LE	<i>Light Edition</i>
LED	<i>Light Emitting Diode</i>

LEP	<i>Light Emitting Polymers</i>
MB	<i>Mega Byte</i>
mm	Milímetro
ms	Milisegundo
NIO	<i>New Input / Output</i>
OLED	<i>Organic Light Emitting Diodes</i>
OPENCV	<i>Open Source Computer Vision Library</i>
PC	<i>Personal Computer</i>
PIP	<i>Personal Interaction Panel</i>
PSP	<i>PlayStation Portable</i>
RAM	<i>Random Access Memory</i>
RF	Radio Frequência
RFID	<i>Radio Frequency Identifier</i>
RFIDG	<i>Radio Frequency Identity and Geometry</i>
RGB	<i>Red, Green, Blue</i>
RIS	<i>Robotics Invention System</i>
RPG	<i>Role-Playing Game</i>
RS-232	<i>Recommended Standard 232</i>
SAR	<i>Spatial Augmented Reality</i>
SW	<i>Software</i>
TAR	<i>Tangible Augmented Reality</i>
TUI	<i>Tangible User Interface</i>
TV	Televisor
UP	<i>Unified Process</i>
USB	<i>Universal Serial Bus</i>
USP	Universidade de São Paulo

VGA	<i>Video Graphics Array</i>
VR	<i>Virtual Reality</i>
XGA	<i>Extended Graphics Array</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	17
1.1 OBJETIVOS	18
1.2 METODOLOGIA	19
1.3 ORGANIZAÇÃO DA DISSERTAÇÃO	20
2 REALIDADE AUMENTADA ESPACIAL	21
2.1 CONCEITOS	21
2.2 DISPOSITIVOS DE EXIBIÇÃO ESPACIAIS.....	24
2.2.1 <i>Video See-Through Baseados em Monitores</i>	24
2.2.2 <i>Optical See-Through Espaciais</i>	25
2.2.3 <i>Dispositivos Espaciais Baseados em Projeções</i>	26
2.3 PROJEÇÕES GEOMÉTRICAS	28
2.4 NOVOS PARADIGMAS DE INTERAÇÃO.....	31
2.4.1 <i>Interfaces Tangíveis</i>	31
2.4.2 <i>Interfaces Horizontais</i>	33
2.5 PROBLEMAS ATUAIS E NOVAS TECNOLOGIAS.....	34
2.6 CONSIDERAÇÕES FINAIS	36
3 JOGOS ELETRÔNICOS.....	37
3.1 CONCEITOS	37
3.1.1 <i>Definição de Jogo</i>	37
3.1.2 <i>Jogos Eletrônicos</i>	38
3.2 TECNOLOGIA.....	40
3.2.1 <i>Plataformas</i>	40
3.2.2 <i>Game Engines</i>	41
3.3 JOGOS APLICADOS EM OUTRAS ÁREAS	42
3.4 EVOLUÇÃO DAS INTERFACES DE JOGOS ELETRÔNICOS.....	44
3.5 JOGOS COM REALIDADE AUMENTADA.....	45
3.5.1 <i>Classificação de Jogos com Realidade Aumentada</i>	47
3.6 CONSIDERAÇÕES FINAIS	48

4 TRABALHOS RELACIONADOS.....	49
4.1 ENJINE.....	49
4.2 ENJINE + ARTOOLKIT	52
4.3 STUDIERSTUBE	54
4.4 DWARF.....	55
4.5 KOBITO	56
4.6 AUGMENTED COLISEUM	57
4.7 CONSIDERAÇÕES FINAIS.....	58
5 DESENVOLVIMENTO DA ROBOT ARENA.....	60
5.1 VISÃO GERAL.....	60
5.2 ARQUITETURA	61
5.3 IMPLEMENTAÇÃO	64
5.3.1 <i>Tecnologias Utilizadas</i>	64
5.3.2 <i>Parte Física</i>	67
5.3.3 <i>Software</i>	72
5.4 PROVA DE CONCEITO E TESTES.....	78
5.5 ANÁLISE DE RESULTADOS E CONSIDERAÇÕES FINAIS.....	80
6 PROTÓTIPOS DE JOGOS DESENVOLVIDOS	82
6.1 FOOTBOT ARENA	82
6.2 TANSPACE	84
6.3 ANÁLISE DOS RESULTADOS E CONSIDERAÇÕES FINAIS.....	85
7 CONCLUSÃO	87
7.1 TRABALHOS FUTUROS	88
REFERÊNCIAS.....	90
APÊNDICE A – DOCUMENTAÇÃO DO PROJETO.....	102

1 INTRODUÇÃO

Nos dias atuais os Jogos Eletrônicos possuem uma grande importância na área econômica e na evolução de diversas áreas da computação. No setor econômico, Smith (2007) estima que o lucro gerado por esta indústria ultrapasse a marca de US\$ 10 bilhões, o que faz deste grande e lucrativo mercado um grande impulsionador das indústrias de *hardware* e *software* computacionais.

Starner et al. (2000) afirmam que jogos são uma plataforma ideal para introduzir novas técnicas de interação homem-máquina, trabalhos colaborativos, agentes inteligentes, gráficos e tecnologias sensoriais. Devido a sua popularidade e a sua natureza lúdica, interagir com jogos faz com que os usuários fiquem mais dispostos a lidar com novas metáforas e tecnologias que ainda não tenham atingido suas características finais.

Esta característica de inovação presente no desenvolvimento de jogos fez com que inúmeras de suas tecnologias se tornassem padrões na indústria de computação, tais como: interfaces gráficas, renderização em tempo-real, antropomorfismo, computação colaborativa, gráficos tridimensionais, *joysticks* e *feedback* sonoro (BUSHNELL, 1996).

No entanto, Magerkurth; Engelke e Memisoglu (2004) relatam que, enquanto a tecnologia dos jogos avança rapidamente, inovações e avanços nos conceitos e conteúdos dos jogos atuais não acompanham esta evolução. Atualmente existe uma insatisfatória convergência entre os jogos eletrônicos lançados no mercado, com a maioria dos títulos se encaixando em poucas e bem definidas categorias, como os jogos de tiro em primeira pessoa e os de estratégia em tempo real. Isto indica que o paradigma atual de jogos atingiu um estágio maduro, onde inovações verdadeiras na forma de jogar são cada vez mais difíceis de se alcançar. Ainda de acordo com esses autores, a aplicação de Realidade Aumentada, que mistura elementos reais e virtuais, é uma das formas de estender estes paradigmas e permitir inovações na jogabilidade, através de diferentes formas de visualização e interação.

1.1 OBJETIVOS

O objetivo desta dissertação é descrever a concepção e desenvolvimento de uma infra-estrutura, denominada Robot ARena, destinada a ser usada tanto como uma plataforma para o desenvolvimento de jogos *indoor* inovadores, quanto em aplicações interativas em geral baseadas em Realidade Aumentada Espacial. Tal plataforma poderá ser usada para explorar novas formas de interação baseadas em projeções de realidade aumentada e interações com robôs (reais e virtuais). Também poderá ser adaptada para aplicações educacionais, como por exemplo, competições de robôs criados e programados por alunos.

Baseada em interfaces horizontais, onde as interações são realizadas sobre a superfície de uma mesa, e composta por elementos de *hardware* e *software*, a Robot ARena, através de uma arquitetura que possibilita diversas formas de interação e de visualização, pode implementar configurações que utilizem rastreamento de objetos, reconhecimento de gestos, superfície sensorial multi-toques, sistemas de projeção superior e retro-projeção. Outro aspecto previsto no projeto da Robot ARena é o controle e a comunicação de robôs, que podem ser utilizados como avatares nos jogos. Esse controle permite uma interação bidirecional entre robôs e as imagens projetadas na arena, de tal forma que não somente os elementos reais interfiram nos virtuais, mas também, os elementos virtuais possam se chocar com os objetos reais tele-controlados.

Como prova de conceito foi implementado um protótipo da Robot Arena com as seguintes características: mesa com visualização de elementos de realidade aumentada por meio de retro-projeção, um sistema de rastreamento de objetos e o controle e comunicação de um robô Lego (LEGO, 2007).

A partir da configuração desta prova de conceito foram desenvolvidos os seguintes protótipos de jogos, que exploram as características desta implementação:

FootBot Arena: jogo que explora as interações entre os elementos reais e virtuais e tem, ainda, como foco a visualização do conteúdo virtual;

TanSpace: implementa o uso de interfaces tangíveis, onde a manipulação de um elemento físico é utilizada como forma de entrada do jogador;

Resumindo, os objetivos deste trabalho são:

- Proposta de uma infra-estrutura (*hardware* e *software*), com as seguintes características:
 - interface horizontal;
 - aplicação de Realidade Aumentada Espacial;
 - interação com robôs tele-controlados;
 - foco em jogos eletrônicos e aplicações educacionais;
 - arquitetura flexível para diferentes aplicações interativas;
 - arquitetura expansível para diferentes formas de interação e visualização.
- Implementação de um protótipo dessa infraestrutura com mesa com visualização de elementos de realidade aumentada por meio de retro-projeção, um sistema de rastreamento de objetos e o controle e comunicação de um robô Lego;
- Desenvolvimento de protótipos de jogos com as seguintes características:
 - Interação entre elementos reais e virtuais;
 - Interface tangível;
 - possibilidade de aplicação em atividades educacionais.

1.2 METODOLOGIA

Como etapa inicial desta pesquisa, foi realizada uma ampla revisão bibliográfica dos principais conceitos e fundamentos de Realidade Aumentada e Realidade Aumentada Espacial, levantando-se técnicas e ferramentas, identificando-se os principais problemas, aplicações e infra-estruturas que facilitem o desenvolvimento de aplicações, em especial aquelas voltadas ao entretenimento e à educação. Também foi objeto desta revisão o estado da arte do desenvolvimento de jogos eletrônicos e de sua cultura, com um principal interesse na aplicação de técnicas de Realidade Aumentada para seu desenvolvimento.

Com a análise desta revisão foi possível estabelecer os objetivos iniciais do projeto e estabelecer uma prova de conceito que demonstrasse a viabilidade e aplicabilidade da infra-estrutura e das técnicas empregadas.

Como metodologia de desenvolvimento de software foi aplicado o Processo Unificado (JACOBSON; BOOCH; RUMBAUGH, 1999), para explorar ao máximo o caráter iterativo e evolutivo do projeto. Um dos principais fatores adotados para contribuir com este aspecto evolutivo é o desenvolvimento de protótipos de jogos que trazem novos desafios para a infra-estrutura.

Por fim, foi realizada uma análise dos resultados do desenvolvimento e testes da infra-estrutura e de cada jogo.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

No segundo capítulo são apresentados os conceitos e tecnologias da primeira parte da revisão bibliográfica envolvida neste trabalho, a **Realidade Aumentada Espacial**.

O terceiro capítulo apresenta a segunda parte da revisão bibliográfica, que tem como objeto os **Jogos Eletrônicos**, sua definição, tecnologia e a aplicação de Realidade Aumentada em seu desenvolvimento.

O quarto capítulo discute os **Trabalhos Relacionados** ao aqui apresentado, com uma ênfase nas infra-estruturas para o desenvolvimento de aplicações com Realidade Aumentada.

O projeto e **Desenvolvimento da Robot ARena** são tratados no capítulo 5, apresentando sua proposta, arquitetura e implementação. Neste capítulo também são discutidos a prova de conceito e os resultados obtidos.

O capítulo seis detalha os **Protótipos de Jogos Desenvolvidos**, seus objetivos e resultados.

No sétimo e último capítulo, são analisados e discutidos os resultados obtidos no desenvolvimento da infra-estrutura e da sua aplicação para o desenvolvimento dos jogos propostos. Também neste capítulo são discutidas futuras pesquisas e possíveis aplicações da Robot ARena, chegando-se à **Conclusão** do trabalho.

2 REALIDADE AUMENTADA ESPACIAL

Neste capítulo serão apresentados os conceitos básicos da Realidade Aumentada Espacial, os novos tipos de dispositivos de exibição que utiliza, os novos tipos de interações que possibilita, seus problemas atuais e as tecnologias emergentes relacionadas que estão sendo pesquisadas.

2.1 CONCEITOS

A Realidade Aumentada (AR, do termo original em inglês *Augmented Reality*) está relacionada à Realidade Virtual (VR, de *Virtual Reality*), mas difere desta, por não ter como objetivo substituir totalmente o ambiente real, e sim acrescentar, em tempo real, elementos virtuais controlados por computadores, que parecem coexistir neste ambiente.

A figura 1 apresenta um exemplo deste conceito, onde o robô real, sobre uma mesa, pode interagir com os elementos virtuais, três baterias e um barril, inseridos na cena.

Figura 1 - Realidade Aumentada, robô real interagindo com baterias e barril virtuais.

Diversas definições formais de AR podem ser encontradas na literatura, Kirner e Tori (2006) listam algumas. A definição que adotada nesta dissertação é a de Azuma et al. (2001) que determina três propriedades para obter-se um ambiente com AR:

- Combinar os elementos reais e virtuais em um ambiente real;
- Permitir interações em tempo real;
- Registrar (alinhar), entre si, os elementos reais e virtuais.

Esses elementos virtuais, embora comumente visuais, não estão restritos somente ao sentido da visão. O aumento da percepção por qualquer sentido, como o tato ou a audição, também pode ser considerado como AR, desde que respeitados os três princípios acima (AZUMA, 1997).

O conceito de AR está presente no contínuo Realidade – Virtualidade, proposto por Milgram et al. (1994) e representado na figura 2, que possui como extremos o ambiente totalmente real e o ambiente totalmente virtual. Neste contínuo, a mistura entre estes dois ambientes é definida como Realidade Misturada, que generaliza a Realidade Aumentada (predominância do real) e Virtualidade Aumentada (predominância do virtual). Estes termos são facilmente compreendidos, porém, em situações limítrofes, a diferenciação entre os conceitos de Realidade Aumentada e Virtualidade Aumentada pode não ser óbvia. Assim muitos autores preferem utilizar apenas um termo para representar qualquer nível de mistura de real e virtual, em geral “Realidade Aumentada” ou “Realidade Misturada”. Neste trabalho foi adotado o primeiro.

Figura 2 - O contínuo Realidade – Virtualidade, adaptada de (MILGRAM et al., 1994).

As três propriedades determinadas por Azuma et al. (2001) estão diretamente relacionadas aos três elementos fundamentais da AR descritos por Bimber e Raskar (2005):

- Rastreamento e registro: determinar de maneira rápida, precisa e robusta o posicionamento do usuário, seu ponto de vista, e a posição de objetos reais e virtuais no ambiente (rastreamento); e buscar alinhar os sistemas de coordenadas do mundo real e do mundo virtual, determinando um ponto de origem para ambos sistemas (registro);
- Tecnologias de dispositivos de exibição: possibilitam aos usuários a visualização do mundo real acrescido de elementos virtuais;
- Renderização em tempo-real: possibilitar a obtenção de cenas misturadas com uma taxa de atualização que permita interações em tempo-real.

Realidade Aumentada Espacial (SAR, do termo *Spatial Augmented Reality*) é uma nova abordagem de AR, apresentada por Raskar; Welch e Fuchs (1998), que traz inovações principalmente nas tecnologias de dispositivos de exibição, mas que influencia diretamente os outros dois elementos. Na SAR, os dispositivos de exibição passam a ser espaciais, ou seja, os elementos virtuais são inseridos diretamente no ambiente físico, principalmente através de projeções (BIMBER; RASKAR, 2005), seguindo uma tendência de evolução natural que a VR já vinha sofrendo (BRYSON et al., 1996).

A figura 3 mostra uma classificação dos tipos de dispositivos de exibição aplicados na AR. Os do primeiro tipo, acoplados ao usuário, são dispositivos normalmente utilizados acoplados à cabeça do usuário, *head-attached displays*. Os dispositivos portáteis são pequenos e leves o suficiente para que sejam carregados na mão dos usuários. Com a tecnologia dos dispositivos espaciais é possível a eliminação dos dispositivos acoplados aos usuários, como os HMDs (*Head-Mounted Displays*), comumente aplicados em AR (KIRNER; TORI, 2006).

Figura 3 - Classificação dos Dispositivos de Exibição, adaptada de (BIMBER; RASKAR, 2005).

2.2 DISPOSITIVOS DE EXIBIÇÃO ESPACIAIS

Os dispositivos de exibição espaciais são divididos em três categorias: *video see-through* baseados em monitores, *optical see-through* espaciais e dispositivos espaciais baseados em projeções. O principal objetivo dessa classe de dispositivos é a substituição de equipamentos acoplados ao corpo do usuário, gerando-se as imagens virtuais diretamente sobre o ambiente (BIMBER; RASKAR, 2005).

2.2.1 Video See-Through Baseados em Monitores

Estes dispositivos utilizam técnicas para misturar o conteúdo virtual com o vídeo real capturado por câmeras e exibir este novo conteúdo aumentado em monitores de vídeo. Comumente estes tipos de dispositivos são chamados de "*Window on the World*" (FEINER et al., 1993). Um exemplo de aplicação deste tipo de dispositivo é mostrado na figura 4, onde é possível ver, somente através do monitor, três

personagens virtuais interagindo com os objetos reais que estão sobre a mesa (AOKI et al., 2005).

Figura 4 - Exemplo de dispositivo “*Window in the World*” aplicado no projeto Kobito (AOKI et al., 2005).

Devido primariamente à limitação do campo de visão permitido pelos monitores, seguido pelo alinhamento com o usuário e a sua distância, o nível de imersão com esta técnica de exibição é normalmente baixo, uma vez que estes limitantes impedem que todo o campo de visão do usuário seja acrescido de elementos virtuais. Outro limitante é a baixa resolução do conteúdo misturado, principalmente do vídeo capturado, que depende da qualidade da câmera utilizada.

2.2.2 Optical See-Through Espaciais

Assim como o *optical see-through* baseado em HMDs (KIRNER; TORI, 2006), a tecnologia dos dispositivos *optical see-through* espaciais baseia-se em componentes óticos, como espelhos, telas transparentes, projetores etc. A principal diferença está no fato de que, no segundo caso, os dispositivos não são acoplados ao corpo do usuário, e as imagens geradas são registradas diretamente sobre o ambiente real. A figura 5 mostra uma aplicação com este tipo de dispositivo de exibição, uma tela semitransparente onde é projetado um vídeo alinhado com o ambiente real.

Figura 5 - Exemplo de aplicação do dispositivo *Optical See-through* Espacial, extraída de (LASER, 2007).

Estes dispositivos permitem uma melhor acomodação dos olhos, uma resolução maior e escalável, assim como é maior o seu campo visual.

Os principais problemas, apresentados por Bimber e Raskar (2005), no uso de tais dispositivos são:

- Não possibilitam aplicações móveis, uma vez que os dispositivos estão alinhados ao ambiente;
- Não possibilitam a manipulação direta do conteúdo virtual e muitas vezes até do real, quando estes se encontram atrás dos dispositivos ópticos.

2.2.3 Dispositivos Espaciais Baseados em Projeções

Os dispositivos espaciais baseados em projeções têm como característica projetar o conteúdo virtual diretamente na superfície de objetos físicos. Esses dispositivos podem ser compostos de um ou de diversos projetores, sendo que o uso de diversos projetores pode aumentar a resolução das imagens projetadas ou aumentar a área de projeção. Na figura 6 é apresentado um conjunto de paredes brancas sem nenhuma projeção e logo abaixo as mesmas paredes com as texturas projetadas de acordo com um modelo virtual.

Figura 6 - Exemplo de aplicação um Dispositivo Espacial Baseado em Projeção, extraída de (LOW et al., 2001).

Quando projeções são utilizadas diretamente sobre um objeto físico tridimensional, para, por exemplo, mudar sua cor ou textura, o próprio objeto acrescenta às projeções características estereoscópicas. No caso destas projeções serem realizadas para representar objetos tridimensionais em superfícies planas, ou que não apresentem a similaridade física do objeto desejado, técnicas de projeções estereoscópicas podem ser incluídas para aumentar a percepção tridimensional do conteúdo virtual. Neste caso, o ponto de vista do observador deverá, obrigatoriamente, ser rastreado.

Assim como os dispositivos espaciais com *optical see-through*, os dispositivos baseados em projeções possuem fatores ergonômicos melhores, resolução escalável, melhor acomodação dos olhos e ainda podem implementar um campo de visão teoricamente ilimitado (de acordo com o número de projetores). Porém estes dispositivos trazem novos problemas, como:

- sombras no conteúdo projetado (pode ser resolvido com múltiplos projetores);

- superfícies que não podem ser usadas para a projeção (limitação de cor, formato e tamanho);
- projetores só podem ter um ponto de foco;
- quanto maior o número de projetores, maior a complexidade de alinhamento e calibração de cores.

2.3 PROJEÇÕES GEOMÉTRICAS

Normalmente, algoritmos de renderização gráfica para dispositivos como monitores de LCD ou CRT podem ser aplicados diretamente para projeções. Porém, estas projeções e superfícies de exibição podem apresentar diversas configurações geométricas. Nesta seção são apresentadas algumas técnicas para renderizar imagens que podem utilizar estas variações, descritas por Bimber e Raskar (2005).

Existem diversas possibilidades que devem ser consideradas para resolver o problema de se renderizar imagens de objetos virtuais tridimensionais utilizando-se um projetor. O usuário pode estar em movimento ou estático, o projetor pode estar iluminando a superfície partindo de qualquer posição, inclusive utilizando uma configuração de projeção frontal ou retro-projeção. Estas superfícies, ainda, podem ser planares ou não planares.

A primeira etapa para a resolução deste problema é estabelecer a relação geométrica entre os componentes envolvidos. A figura 7 mostra os componentes e suas relações.

Figura 7 - Relação entre componentes envolvidos na projeção geométrica, adaptada de (BIMBER, RASKAR, 2005).

De acordo com a figura, é estabelecida uma relação entre o usuário rastreado T , o projetor P , caracterizado por seu centro de projeção P_c e pelo seu plano de imagem P_r , e a superfície de exibição D . Assim, para qualquer ponto tridimensional do objeto virtual V , é possível expressar o mapeamento entre V e o *pixel* do projetor m_p , na imagem renderizada do objeto. A projeção m_p de um ponto V é a intersecção do raio P_cM com o plano de imagem do projetor P_r , onde M é a intersecção do raio TV com a superfície de exibição D .

Nesta relação geométrica estão presentes três componentes, que definem a configuração do dispositivo de exibição:

- Modelo do Projetor: o modelo do projetor pode ser aproximado utilizando-se um modelo de projeção *pin-hole*, similar ao modelo de câmera de orifício (*pin-hole*), assim a geração da imagem pode ser obtida através da matriz de projeção presente nas aplicações gráficas tradicionais;
- Portal de Exibição: é a superfície de exibição, irregular ou não, pode ser representada utilizando uma malha poligonal com forma aproximada;
- Posição do Usuário: a posição do usuário é representada simplesmente por um ponto tridimensional no sistema de coordenadas do mundo. Este ponto de vista pode ser rastreado; É necessário um ponto de vista para cada olho, para projeções estereoscópicas.

Determinados os componentes e suas relações geométricas, é possível chegar a uma estratégia de renderização para encontrar a imagem que o projetor deve gerar, e iluminar corretamente a superfície de exibição. Esta estratégia é dividida em dois passos:

- Primeiro, no campo de projeção do ponto de vista desejado, é representado o raio $\mathbf{TV}$ em um plano de projeção arbitrário. Isto é equivalente a calcular a imagem de $\mathbf{V}$ da posição do usuário $\mathbf{T}$ naquele plano de projeção. Esta projeção é determinada pela matriz de projeção $\mathbf{P}_T$ e o *pixel* projetado pelo $\mathbf{m}_T$ no plano da imagem Π_T . O *pixel* $\mathbf{m}_T$ é calculado (*shaded*) de acordo com o ponto do objeto mais próximo sobre o raio $\mathbf{TV}$. O *pixel* $\mathbf{m}_T$ é transferido para o ponto mais próximo no portal de exibição seguindo o raio $\mathbf{Tm}_T$. O portal de exibição é aumentado com a imagem transferida de Π_T através do centro da projeção $\mathbf{T}$; Este processo assemelha-se à aplicação de uma textura no modelo da superfície de exibição partindo do ponto de vista do usuário;
- No segundo passo é encontrada a imagem do portal de exibição aumentado; Considerando que a matriz de projeção $\mathbf{P}_P$ determina os parâmetros intrínsecos e extrínsecos do projetor (modelo e posição do projetor), esta matriz é utilizada para renderizar a imagem do portal de exibição aumentado, assim, o ponto $\mathbf{M}$ é transferido para o *pixel* $\mathbf{m}_P$.

A representação destes dois passos aplicados na estratégia de renderização pode ser vista na figura 8.

Figura 8 - Estratégia de renderização para projeção geométrica, adaptada de (BIMBER; RASKAR, 2005).

2.4 NOVOS PARADIGMAS DE INTERAÇÃO

Raskar e Low (2001) afirmam que a interface entre um ser humano e um objeto físico é a essência da interação intuitiva, uma vez que não existem *widgets*, janelas ou monitores para manipulá-lo. Para interagir com este objeto pode-se andar à sua volta, aproximar ou afastar, olhar especificamente para um ponto de interesse, tocá-lo, modificá-lo com as mãos ou com algum tipo de ferramenta, tudo isso com um alto grau de acuidade visual, espacial e temporal. Por outro lado, as interfaces digitais também oferecem inúmeras vantagens e confortos, como uma rápida operação de “desfazer”, resolução de cálculos complexos com um simples clique, automação de tarefas tediosas e repetitivas etc.

Um dos principais benefícios que as interfaces de SAR oferecem é a união da simplicidade das interfaces naturais com o poder dos recursos computacionais. Na SAR os objetos reais e virtuais estão intimamente ligados, muitas vezes o próprio objeto real pode ser a superfície, ou a “casca” do objeto virtual, portanto, as interfaces aplicadas neste ambiente devem explorar ao máximo essa proximidade entre os elementos, e aproveitar a familiaridade dos usuários com as interfaces do mundo real.

O aspecto principal que influencia a elaboração destas interfaces mais naturais é a eliminação da interação direta com os computadores, ou o uso de computadores e equipamentos acoplados aos usuários. Naturalmente a SAR segue este aspecto, que pode ainda ser reforçado com o uso das interfaces tangíveis.

2.4.1 Interfaces Tangíveis

A Interface Tangível (TUI, *Tangible User Interface*) foi definida por Ishii e Ullmer (1997) como uma interface homem-máquina que utiliza as interações com o mundo físico como metáforas para manipulação do mundo virtual. Desta forma um elemento físico e um elemento virtual são associados, e quando ocorre alguma mudança em um deles, normalmente no objeto físico, o outro é afetado da mesma maneira,

permitindo que o usuário manipule um objeto virtual em um ambiente 3D através da manipulação de um objeto real, de uma maneira muito mais intuitiva.

Através da TUI a manipulação de conteúdos digitais se torna muito mais natural, como por exemplo: rotacionar um cubo real e obter as mesmas transformações em um cubo virtual em um ambiente tridimensional. Este tipo de interação pode substituir uma interface digital que se baseie no *mouse* para selecionar o objeto virtual, evitando assim a necessidade de metáforas complexas para realizar transformações em seus três eixos. Um exemplo de aplicação da interface tangível é apresentado por Underkoffler e Ishii (1998), em uma aplicação chamada *Illuminating Light*, onde os usuários podem manipular objetos físicos que representam componentes óticos (espelhos, *lasers*, lentes, etc) em uma superfície aumentada. Quando o usuário posiciona o objeto que representa o *laser* nesta superfície, o sistema rastreia sua posição e projeta um raio luminoso que aparenta sair diretamente deste objeto. Outros objetos podem ser posicionados neste ambiente, como o objeto que representa um espelho, que faz com que o raio seja refletido de acordo com sua posição, e assim é possível configurar um arranjo de elementos óticos para obter um efeito desejado.

A colaboração entre os usuários é permitida e até mesmo incentivada com o uso da TUI. Isto ocorre devido ao fato de as interações e movimentos ocorrerem mais livremente, com menos limitadores. Como exemplo desta liberdade pode ser citada a interação com as duas mãos e até mesmo com o corpo. Outro fator que beneficia a colaboração é o fato de ser possível saber quem é o responsável por um objeto virtual, simplesmente observando quem está manipulando sua representação real.

Brave et al. (1998) fazem uso da TUI em aplicações colaborativas com o auxílio de computadores com comunicação remota. Uma dessas aplicações cria a ilusão de que usuários remotos estão manipulando o mesmo objeto físico, que representa algum conteúdo virtual. Este efeito é conseguido através de um sistema que sincroniza a posição real do objeto de acordo com a manipulação de cada usuário remoto, que parecem estar interagindo com o mesmo objeto.

Mais intimamente relacionada com a AR, Billinghurst et al. (2005) definem a Realidade Aumentada Tangível (TAR, *Tangible Augmented Reality*), baseada nos mesmos princípios da TUI, e concluem que, para o uso das interfaces tangíveis ser intuitivo e natural, sem a necessidade de um treinamento específico para sua

utilização, é necessário a escolha de objetos físicos e metáforas comuns aos usuários que irão utilizá-las. Assim os usuários podem basear-se nas suas habilidades e experiências desenvolvidas durante o curso de sua vida.

2.4.2 Interfaces Horizontais

Interfaces horizontais, ou interfaces *tabletop*, são interfaces que utilizam uma mesa ou uma superfície horizontal como suporte para uma interface homem-máquina. Mesas oferecem um amplo espaço físico conveniente para reuniões, conversas, análises de documentos e outras atividades que necessitem de uma colaboração face a face (SHEN et al., 2004). Já os computadores atuais, como *desktops*, *notebooks* e *handhelds*, lidam muito bem com informações digitais, porém com um foco muito grande em interações individuais. As interfaces horizontais integram as facilidades de um computador sobre a superfície de uma mesa, permitindo interações colaborativas e tangíveis, para isso emprega sistemas de multi-toques, reconhecimento de gestos e detecção e rastreamento de objetos (WU; BALAKRISHNAN, 2003).

Um exemplo de interface horizontal é o PlayAnywhere (WILSON, 2005) um dispositivo composto por um projetor, uma câmera e uma fonte de iluminação infravermelho, que tem como objetivo transformar qualquer superfície em uma interface horizontal, ou uma superfície aumentada. Baseada em visão computacional, reconhece os toques na superfície através da identificação da sombra do dedo indicador dos usuários (só permite um ponto de toque por mão), reconhece a posição, orientação e tamanho de folhas de papel sobre a superfície, permitindo que conteúdos virtuais sejam projetados sobre estas folhas, disponibiliza um conjunto de marcadores fiduciais para que outros objetos possam ser inseridos, também implementa um algoritmo baseado em *optical flow* para a manipulação de objetos virtuais na superfície aumentada.

Outro aspecto importante das interfaces horizontais é o *software* que possibilita o tratamento e o gerenciamento das interações e visualizações que este tipo de interface apresenta, ou seja, oferecer uma interatividade em tempo real de múltiplos e simultâneos eventos de entrada e múltiplas manipulações de objetos concorrentes.

Shen et al. (2004) desenvolveram o DiamondSpin um *toolkit*, em Java, que auxilia na criação de interações colaborativas ao redor de uma mesa. DiamondSpin permite que documentos digitais sejam posicionados livremente, suporta múltiplas áreas de trabalho na mesma superfície aumentada, operações multi-usuários através de eventos de entrada *multi-thread*, múltiplos objetos ativos e múltiplos menus concorrentes. O DiamondSpin pode, ainda, gerenciar facilmente superfícies poligonais e circulares.

2.5 PROBLEMAS ATUAIS E NOVAS TECNOLOGIAS

Alguns dos principais problemas presentes nos dispositivos de exibição espaciais atuais são: pouca mobilidade, devido ao tamanho dos projetores e utilização de dispositivos fixos no ambiente, dificuldades em se projetar imagens em certos tipos de superfícies e interferência da iluminação ambiente (BIMBER; RASKAR, 2006).

Esses problemas serão solucionados com as novas gerações de projetores, que serão mais compactos e leves. Esses projetores ainda terão tecnologia que possibilita a projeção de imagens em alta frequência, com um alto brilho e capacidade de foco quase infinito (BIMBER; RASKAR, 2006).

Outras tecnologias para dispositivos de exibição espaciais, que não projetores, também estão sendo desenvolvidas, como, por exemplo, os dispositivos de exibição flexíveis. Uma dessas possibilidades é o uso de Diodos Emissores de Luz Orgânicos (OLEDs, *Organic Light Emitting Diodes*) (HOWARD; PRACHE, 2001), que poderão permitir o desenvolvimento de telas com um baixo custo e alta definição. A principal vantagem dos OLEDs é seu tamanho molecular, que permite que estes sejam depositados em substâncias líquidas, e depois “pintados” em uma superfície que será o dispositivo de exibição. Outras possibilidades são os LEPs (*Light Emitting Polymers*) (BURROUGHES et al., 1990) e os *Electronic Papers* (XEROX, 2007).

Algumas técnicas que estão sendo pesquisadas são capazes de gerar *fótons* visíveis dentro de materiais de suporte transparentes, excitando-se opticamente *íons* ativos através de fontes de energias especiais, normalmente através do cruzamento dessas energias, como lasers infravermelhos, fontes de radiação ultravioleta ou feixes de *elétrons*, em um certo ponto tridimensional, criando-se assim um *voxel* no

espaço (DOWNING et al., 1996). A possibilidade de usar-se o ar, sem outras substâncias, como esse material de suporte representaria a tecnologia ideal, não só para os dispositivos de exibição de SAR, mas para a tecnologia de exibição tridimensional como um todo (BIMBER, RASKAR, 2005).

Não são somente as técnicas de exibição que apresentam problemas e novas possibilidades, um dos principais problemas na área de rastreamento e determinação de posição é a influência do ambiente, como cor e iluminação, no reconhecimento de marcadores visuais. Outro problema é que nem sempre é prático adicionarem-se marcadores visuais distintos em um mesmo ambiente. Marcadores eletrônicos possuem inúmeras vantagens sobre os marcadores visuais, como a possibilidade de existirem vários marcadores em um mesmo local, não serem sensíveis à luz e não precisarem estar no campo de visão da câmera ou do usuário para serem detectados. Normalmente estes marcadores são baseados em técnicas de rádio frequência, ultra-som ou dispositivos magnéticos.

Novas abordagens para estes marcadores estão surgindo, como a proposta do RFIG (*Radio Frequency Identity and Geometry*) (RASKAR et al., 2004), que consiste do acréscimo de um foto-sensor em *tags* de RFID (*Radio-Frequency Identifier*). O RFIG é aplicado em sistemas com AR, onde um dispositivo portátil com um projetor é carregado pelo usuário. Este dispositivo basicamente é composto, além do projetor, de uma câmera e um *transponder* de RF.

Os *tags* de RFIG são colocados em objetos que serão identificados. Em uma primeira etapa o dispositivo portátil do usuário ativa estes RFIGs que estão em sua área de cobertura, e projeta uma luminosidade sobre seu campo de visão, os RFIGs enviam suas informações e indicam se estão no campo visível do projetor. Na próxima etapa é projetada sobre os objetos, que possuem os marcadores, uma sequência de padrões de luminosidade que faz com que cada *pixel* do projetor emita um único e temporal código-Gray. O foto-sensor grava este código e envia para o dispositivo móvel. Este por sua vez calcula quais os *pixels* projetados atingiram cada RFIG, possibilitando que o conteúdo aumentado possa ser corretamente projetado sobre os objetos com os *tags*.

2.6 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os conceitos de Realidade Aumentada (AR) e Realidade Aumentada Espacial (SAR), esta última sendo uma das principais características da Robot ARena, misturando elementos reais com elementos virtuais. A SAR possibilita três dispositivos de exibição, a infra-estrutura, desenvolvida neste trabalho, implementa os Dispositivos de Exibição Baseados em Projeções, tanto com projeções frontais (ou superiores) como com retro-projeções. A aplicação deste tipo de dispositivo ainda permite a utilização de projeções geométricas para a visualização do conteúdo aumentado.

Foram também apresentados os novos paradigmas de interação, baseados em SAR, interfaces tangíveis e interfaces horizontais, e os principais problemas relacionados a essas tecnologias atualmente pesquisados.

3 JOGOS ELETRÔNICOS

Este capítulo irá conceituar Jogos e Jogos Eletrônicos, discutir suas principais características, suas aplicações em diversas áreas, inclusive no campo da pesquisa, e as principais tecnologias envolvidas.

Ainda, neste capítulo será apresentada uma breve evolução das interfaces de jogos e serão discutidas novas possibilidades trazidas pela Realidade Aumentada aos jogos eletrônicos.

3.1 CONCEITOS

Conceituar ou definir “jogo” é uma tarefa extremamente difícil, não somente pela amplitude do conceito (HUIZINGA, 2004), mas, também, pela necessidade de sua diferenciação em relação a outras atividades, como brincar ou executar tarefas cotidianas de nossas vidas.

Diversos autores (WALTHER, 2003) (CRAWFORD, 2003) (CAILLOIS, 1958), das mais diferentes áreas, procuram definir “jogo”, mas normalmente se limitam a conclusões adequadas apenas a seus objetivos mais imediatos e que, não raramente, conflitam entre si. Alguns autores simplesmente concluem que é impossível obter-se uma definição única (WITTGENSTEIN, 1999). Na falta de um consenso, neste trabalho foi adotada uma definição que atenda aos objetivos do trabalho, sem pretensão de que seja a mais abrangente.

3.1.1 Definição de Jogo

Salen e Zimmerman (2004), na busca para definir “jogo”, mapearam os elementos presentes em diversas definições. Com esse mapeamento, observa-se que os elementos mais presentes nas definições são as Regras e os Objetivos/Resultados. Além destes elementos, um jogo também deve conter conflitos ou obstáculos, sejam

eles determinados pela estrutura do jogo, por outros jogadores ou por influência da aleatoriedade. Estes elementos são suficientes para determinar o que é um jogo, porém eles podem definir também outras atividades que com certeza não são jogos, como escrever um livro. Para solucionar este problema um outro elemento é adicionado, a artificialidade, resultando na seguinte definição:

“Um Jogo é um sistema no qual jogadores têm a experiência de um conflito artificial, definido por regras, que possibilita um resultado quantitativo.” (SALEN; ZIMMERMAN, 2004).

Os principais elementos desta definição são caracterizados da seguinte forma:

- Jogadores: para haver um jogo é necessário que existam participantes ativos, e que estes participantes interajam com o jogo para obter a experiência de jogar;
- Conflito: um jogo deve conter um desafio, este desafio pode assumir diversas formas, desde competição até cooperação, pode ser um desafio solitário ou um sistema com conflitos sociais entre vários jogadores;
- Regras: é através das regras que a estrutura de um jogo é estabelecida, são elas que permitem que um jogador jogue, delimitando o que podem ou não fazer;
- Resultados: jogos devem ter um objetivo ou resultado; No fim de um jogo o jogador deve saber se ganhou ou perdeu, ou receber a indicação de algum resultado numérico.

3.1.2 Jogos Eletrônicos

Jogos Eletrônicos são jogos que utilizam algum dispositivo eletrônico (como computadores, consoles de *videogame*, celulares etc) como seu principal meio, para, por exemplo, controlar seu estado, armazenar informações, controlar as regras e possibilitar a interação através de diversas formas (TORI et al., 2007). Em uma outra definição, onde jogos de computadores equivalem a jogos eletrônicos, Bernardes Jr. (2006) afirma que: “Jogo de Computador é qualquer jogo que use o computador

explicitamente como meio ou como facilitador, normalmente usando um *software* específico para isso, que contém uma representação desse jogo.”

Jogos desenvolvidos utilizando algum dispositivo eletrônico como meio podem fazer uso e explorar algumas características, que, embora possam ser também aplicadas de alguma maneira em jogos não eletrônicos, em jogos eletrônicos elas são implementadas de forma mais robusta (SALEN; ZIMMERMAN, 2004). Estas características são:

- Interatividade instantânea, porém mais restrita: jogos eletrônicos podem oferecer respostas interativas imediatas às entradas dos jogadores, eles podem oferecer uma jogabilidade que muda e reage dinamicamente às decisões dos jogadores. Apesar desta característica, os tipos de interação que os meios eletrônicos podem interpretar são mais limitados, principalmente se tratando de seus mecanismos de entrada, como por exemplo, o uso do teclado e mouse à frente de todas as possibilidades de interação que jogos não eletrônicos, como o futebol, podem oferecer;
- Manipulação de informações: jogos eletrônicos fazem um ótimo uso e armazenamento de dados digitais, como texto, imagens, vídeo, sons, animações, computação gráfica, etc. Não somente dados digitais, mas também mecânicas e regras do jogo, que permitem que um jogador aprenda a jogar durante o jogo, e também informações sobre o jogo, que podem ser escondidas e apresentadas aos jogadores nos momentos certos e de diferentes formas;
- Automatização de sistemas complexos: uma das maiores vantagens dos jogos eletrônicos é que eles não somente armazenam as regras dos jogos, mas podem resolvê-las rapidamente e de forma transparente aos jogadores, o que possibilita a utilização de sistemas e cálculos complexos, que em jogos não eletrônicos seria difícil implementar;
- Comunicação remota: os dispositivos eletrônicos, além de poderem manipular diversos tipos de informações, podem fazê-lo remotamente, trocando informações, permitindo assim que jogadores, distantes um do outro, possam interagir com o mesmo jogo e interagir entre si, de diversas maneiras, como através de *chats* dentro do jogo ou através da própria jogabilidade oferecida.

3.2 TECNOLOGIA

O desenvolvimento de jogos envolve tecnologias tanto de *hardware* quanto de *software*. Como forma de simplificação, neste capítulo o *hardware* será apresentado como plataformas de desenvolvimento ou simplesmente plataformas, onde os jogos são executados. Como *software* serão discutidos os *game engines*, partes reaproveitáveis no desenvolvimento dos jogos.

3.2.1 Plataformas

Devido à sua popularidade, os jogos eletrônicos estão sendo desenvolvidos usando-se como plataforma diversos tipos de *hardware*, que podem ser dedicados, como consoles, ou mesmo telefones celulares. Bianchini (2005) lista estas plataformas:

- Consoles dedicados, que se conectam a aparelhos de TV: possuem *hardware* dedicado à execução de jogos, como exemplos podem ser citados: Playstation 1, 2 e 3 da Sony, GameCube e Wii da Nintendo e Xbox e Xbox 360 da Microsoft;
- Computadores pessoais: utilizam o *hardware* de PCs, e seus sistemas operacionais como Windows, MacOS e Linux, muitos procuram explorar os aspectos multi-jogadores (comunicação remota) que esta plataforma permite;
- Arcade: máquinas de Fliperama, estas máquinas também possuem *hardware* dedicados, mas também possuem um monitor, ou TV, dedicado;
- Consoles portáteis: dedicados a jogos, os mais comuns são o Nintendo DS da Nintendo e o PSP (*Playstation Portable*) da Sony;
- Televisão interativa: jogos sediados em empresas de televisão, normalmente com conteúdos pagos, controlados pela linha telefônica, antena parabólica ou cabo;

- Telefones celulares e outros dispositivos portáteis: não permitem o uso de gráficos e jogabilidade tão avançadas como as empregadas em PCs e consoles, devido à limitação de seu *hardware*. Mas conforme o avanço destas plataformas, prognósticos mostram que uma grande porcentagem de jogos será desenvolvida para estas plataformas (BIANCHINI, 2005).

3.2.2 Game Engines

Como apontado por Blow (2004), a complexidade dos jogos eletrônicos vem crescendo muito nos últimos anos, o que afeta diretamente não só o tamanho e complexidade dos projetos de desenvolvimento de *software*, mas também o seu número de requisitos específicos.

Jacober (2007) afirma que, como em qualquer *software* que atinja um certo nível de complexidade, deve-se reaproveitar partes, mais genéricas, deste trabalho em novos projetos, acrescentando as especificidades necessárias. No desenvolvimento de jogos, a parte reaproveitada em outros projetos é chamada de *Game Engine*. Este termo também pode ser encontrado, em uma tradução livre, como Motor de Jogo. Neste trabalho foi adotado o termo em inglês, por ser mais difundido, e foi definido como: “um conjunto de componentes de software integráveis e reutilizáveis, que fornecem serviços utilizados em um jogo eletrônico” (BIANCHINI et al., 2006).

A figura 9 mostra a estrutura básica de um jogo, e a integração de um *game engine*, que atua como uma camada entre a plataforma onde o jogo é executado e os elementos e regras aplicadas no jogo.

Figura 9 – *Game engine* na estrutura de um jogo, adaptada de (LEWIS; JACOBSON, 2002).

Um *game engine* pode ser formado por diversos componentes, ou módulos, com diferentes especializações, a figura 10 mostra um conjunto de componentes usualmente encontrados em um *game engine*. Estes componentes são mais bem discutidos em Bianchini et al. (2006).

Figura 10 – Principais componentes de um *Game Engine*.

3.3 JOGOS APLICADOS EM OUTRAS ÁREAS

Segundo Rosenbloom (2003), jogos são envolventes por fornecer ambientes tridimensionais, interativos e imersivos, com visuais realistas, características que reforçam a experiência do jogador. Estes aspectos fazem com que a tecnologia aplicada a jogos desperte interesse em aplicações cujo principal objetivo não é o entretenimento, estas aplicações são normalmente denominadas de “*Serious Games*” (PURDY, 2007).

Swartout e van Lent (2003) identificam duas classes de aplicações que podem se beneficiar da abordagem de jogos para seu desenvolvimento. A primeira classe é chamada de sistemas baseados em experiência, que leva em conta o fato do *design* de jogos ser voltado para a experiência dos jogadores, algumas aplicações que podem aproveitar esta abordagem são: educação e treinamento, comunicação e persuasão e terapia baseada em experiências. A segunda classe utiliza uma abordagem baseada em jogos para criar uma infra-estrutura de teste para tecnologias emergentes. Os mundos virtuais criados para os jogos podem refletir o mundo real, porém possuem muitos aspectos mais simplificados e controlados,

assim, novas tecnologias podem ser testadas em um ambiente comparativamente rico, antes de serem aplicadas ao mundo real.

A utilização de jogos em pesquisas acadêmicas, tanto nas que os utilizam como ferramenta quanto as que possuem jogos como objeto de pesquisa, vem se tornando cada vez mais comum (BIANCHINI et al., 2006). Em Rhyne (2002), a autora discute como as tecnologias desenvolvidas para jogos, devido ao seu rápido avanço, influencia e pode influenciar ainda mais as pesquisas que envolvem Visualização Científica e VR.

Recentemente a indústria também vem adotando os *serious games*. Purdy (2007) expõe alguns exemplos de como e onde estes jogos estão sendo aplicados:

- Educação e Treinamento: jogos estão sendo usados tanto para um aprendizado detalhado e específico, como para um aprendizado de alto nível. Alguns jogos introduzem aos estudantes tópicos de alto nível, capturando sua atenção ao assunto, encorajando-os a aprender mais, outros tipos de jogos são mais profundos e detalham algum assunto, ou desenvolvem alguma habilidade específica, possuindo assim uma maior característica de simulação;
- Recrutamento: empresas que desejam recrutar uma nova geração de funcionários, que cresceram com o convívio regular com jogos, onde suas interações e comunicação já são virtuais. Um exemplo de sucesso deste tipo de aplicação é o jogo *American's Army*, desenvolvido pelo exército americano (ZYDA et al., 2003) que divulga o cotidiano da vida e treinamento das tropas americanas;
- Marketing: recentemente algumas empresas estão utilizando jogos para demonstrar aos clientes que seus produtos e serviços são melhores, ou ainda promover a marca e o nome da empresa. Estes jogos procuram manter o fator de entretenimento, mas sempre girando em torno de mensagens de *marketing*, estes jogos também são conhecidos como *advergames*.

Smith (2007) discute esta crescente adoção de jogos por indústrias bem estabelecidas e, de acordo com sua teoria de impacto dos jogos, identifica cinco forças que induzem as empresas à adoção de *serious games*. Estas forças são:

- Custos do *hardware* de computadores: devido à significativa redução nos custos do *hardware* de computadores, necessários para suportar aplicações baseadas em jogos;
- Sofisticação e poder das aplicações de *software*: o uso de agentes inteligentes, mundos virtuais tridimensionais, interface de usuários acessível e simples, modelos físicos, redes de comunicação globais e mundos persistentes;
- Aceitação social de ferramentas de jogos: crescente aceitação social de soluções baseadas em jogos utilizadas em aplicações convencionais na indústria;
- Sucessos em outras indústrias: treinamento militar, experimentos de química, treinamento corporativo, *design* arquitetônico, etc;
- Experimentação em indústrias inovadoras: surgimento de novas experimentações com a tecnologia dos jogos e a busca por novas maneiras de melhorar práticas já estabelecidas na indústria.

De acordo com o autor (SMITH, 2007), estas cinco forças podem fazer com que a tecnologia de jogos se torne um padrão recomendado para a solução de diversos problemas na indústria.

3.4 EVOLUÇÃO DAS INTERFACES DE JOGOS ELETRÔNICOS

Rollings e Adams (2003) discutem a evolução que as interfaces dos jogos sofreram durante o passar dos anos. O principal fator que atua diretamente na evolução destas interfaces é o avanço tecnológico, sendo que os pontos principais desta evolução serão discutidos aqui.

Como exemplo, os autores analisam os jogos de Aventura que, eram totalmente baseados em textos, no início dos jogos comerciais. O jogador interagia com o jogo lendo uma descrição do ambiente e a situação, e digitando ações em forma de comandos, que normalmente tinham o formato: “PEGAR ITEM” ou “MOVER ESQUERDA”.

Com a evolução tecnológica, foram surgindo novas possibilidades. O primeiro passo foi a melhora do interpretador de texto, permitindo-se a entrada de comandos com sintaxe mais próxima à da linguagem natural. Aos poucos novos elementos foram sendo adicionados, principalmente elementos gráficos, que foram evoluindo até se chegar aos ambientes totalmente tridimensionais, onde a interação é feita através de menus sensíveis ao contexto, possibilitando ao jogador interagir com praticamente qualquer elemento do ambiente do jogo.

O que ocorreu ao longo desse tempo, sem dúvida foi a sofisticação das interfaces, e principalmente dos controles. Com a evolução do *hardware*, os jogos passaram a apresentar uma maior gama de opções, tornaram-se maiores e com mais resolução, exigindo dos jogadores mais ações simultâneas. Para isso os controles passaram a ter mais botões e direcionais analógicos e digitais. Mesmo com esse desenvolvimento a forma de interação permanece praticamente a mesma, apertando botões e direcional, ou através de um teclado e mouse.

O console de *videogame*, recentemente lançado, Nintendo Wii (NINTENDO, 2007), com seu controle sem fio e sensores de movimentos, traz novas possibilidades de interação e interface em seus jogos. Dependendo da sua implementação, este controle pode atuar em alguns jogos como uma interface tangível, como por exemplo no jogo eletrônico “*Wii Sports*”, onde o controle pode ser utilizado como uma raquete de tênis ou como um taco de *baseball*.

Em uma implementação mais direta de AR, a Sony com o seu EyeToy (EYETOY, 2007), para Playstation, implementa um jogo de cartas chamado “*Eye of Judgement*”. Estas cartas são utilizadas como marcadores fiduciais, permitindo o registro e a visualização, na televisão, das criaturas tridimensionais que estas cartas representam, criando assim um ambiente aumentado.

3.5 JOGOS COM REALIDADE AUMENTADA

Embora as interfaces dos jogos tenham evoluído com o passar dos anos, criando-se ambientes graficamente mais sofisticados, e disponibilizando-se controles mais elaborados, a forma de interação com os jogos não sofreu nenhuma mudança

relevante, permanecendo com o uso de controles direcionais e o pressionamento de botões, ou mouse e teclado.

A AR quebra esse paradigma e possibilita novas formas de interação e interfaces para os jogos. A primeira grande mudança na interface do jogo é a visualização, que passa a fazer parte do ambiente real, e não somente em um mundo virtual no monitor.

Como apresentado no capítulo 2, aplicações baseadas em AR demandam que as interfaces sejam mais naturais, utilizando-se para isso as metáforas do mundo real, como interfaces tangíveis, que são intuitivas nessa nova tecnologia para jogos. Permite, também, que ambientes que utilizam AR, tenham um aspecto colaborativo e incentiva a interação direta entre os jogadores, dispensando avatares ou *chats* dentro do jogo, ampliando muito o aspecto social dos jogos. Magerkurth et al. (2005) definem que em jogos eletrônicos tradicionais, em computadores e consoles, não existe esta interação, devido ao fato de suas interações serem centralizadas na tecnologia. Com o uso de AR, o jogo é trazido ao mundo real, as pessoas passam a ter uma interação mais física e pessoal, fortemente emocional, como em jogos de tabuleiros tradicionais.

Com o objetivo de examinar as possibilidades trazidas pela aplicação de AR em jogos, Nilsen; Linton e Looser (2004) caracterizam a experiência dos jogadores em quatro aspectos: físico, mental, social e emocional. Baseados nestes aspectos eles comparam as características dos jogos tradicionais e dos jogos de computadores. A tabela 1 apresenta estas características.

Tabela 1 – Características dos Jogos Tradicionais e Jogos Eletrônicos.

	Físico	Mental	Social	Emocional
Jogos Tradicionais	<ul style="list-style-type: none"> - Jogadores podem utilizar todo o corpo; - O mundo real fornece o ambiente do jogo; - Objetos físicos podem ter significado no jogo. 	<ul style="list-style-type: none"> - Jogadores não querem resolver regras complexas; - Permite o raciocínio espacial, principalmente 3D. 	<ul style="list-style-type: none"> - Permite uma comunicação direta e natural; 	<ul style="list-style-type: none"> - Pode estimular todos os sentidos dos jogadores; - Limitado pela habilidade prática de controlar o ambiente.
Jogos Eletrônicos	<ul style="list-style-type: none"> - Interação física limitada pelos dispositivos de entrada. 	<ul style="list-style-type: none"> - Permite regras e modelos de jogos complexos - Pode disponibilizar oponentes e agentes com Inteligência Artificial. 	<ul style="list-style-type: none"> - Controla a exposição das comunicações; - Permite jogos remotos e com diversos jogadores. 	<ul style="list-style-type: none"> - Potencial para criar diversos ambientes virtuais e cenários; - Limitado ao estímulo visual e auditivo.

(NILSEN; LINTON; LOOSER, 2004)

Nenhum jogo utiliza apenas um destes aspectos, a maioria dos jogos combina todos eles. Enquanto jogos eletrônicos possuem vantagens em algumas características, os jogos tradicionais possuem em outras. A grande vantagem do desenvolvimento de jogos com AR é a possibilidade de aplicar as características presentes em cada meio.

3.5.1 Classificação de Jogos com Realidade Aumentada

Os jogos com AR podem ser divididos em duas classificações: os jogos *outdoor* e os jogos *indoor*. Em Bernardes Jr.; Dias e Tori (2005) estes jogos são classificados primariamente quanto a seu espaço de jogo, enquanto os jogos *outdoors* ocupam espaços amplos e normalmente abertos, os jogos *indoor* são normalmente utilizados em espaços pequenos e fechados.

Os jogos *outdoor* usam, em sua maioria, metáforas físicas de interação, como passar por cima de objetos virtuais, carregá-los, tocar os outros jogadores, correr ou andar em algum espaço amplo. Exemplos desses tipos de jogos e de suas interfaces podem ser vistos em: *Human Pacman* (CHEOK et al., 2004) e *ARQuake* (PIEKARSKY; THOMAS, 2002).

Nos jogos *indoor* também existe a metáfora física de interação, mas em menor amplitude, como a manipulação de objetos em uma mesa, ou o movimento e gestos dos jogadores em uma única sala. Exemplos desses jogos podem ser vistos em *Herding Sheep* (MACWILLIAMS et al., 2003), Magerkurth et al. (2005), Ulbricht e Schmalstieg (2003) e o SENAC ARHockey (VIEIRA et al., 2006) que pode ser visto na figura 11, onde dois jogadores utilizam dispositivos tangíveis para rebater o disco virtual em um ambiente virtual projetado.

Figura 11. O jogo *indoor* SENAC ARHockey, extraída de (VIEIRA et al., 2006).

3.6 CONSIDERAÇÕES FINAIS

O principal foco da Robot ARena é o desenvolvimento de jogos com interfaces baseadas nos novos paradigmas de realidade aumentada e interfaces tangíveis. Neste capítulo foram discutidos o conceito de “jogo” e a tecnologia de jogos eletrônicos. Também foi discutido como as interfaces dos jogos estão evoluindo e como a AR é uma nova fronteira nessa evolução, permitindo a inclusão de características de jogos tradicionais em jogos eletrônicos.

Os jogos com AR são classificados como *outdoor* e *indoor*, a Robô ARena se concentra no desenvolvimento de jogos *indoor*.

Conforme visto, jogos possuem um potencial para ir além de objetivos puramente de entretenimento, podendo ser aplicados em diversas áreas. Assim, a Robot ARena pode explorar essa característica, principalmente facilitando o desenvolvimento de jogos educacionais.

4 TRABALHOS RELACIONADOS

Este capítulo apresenta trabalhos que de alguma forma estão relacionados com o trabalho apresentado nesta dissertação, com especial atenção para *frameworks* e infra-estruturas que auxiliam o desenvolvimento de aplicações com AR.

4.1 ENJINE

O enJine (ENJINE, 2007) (BERNARDES JR; NAKAMURA; TORI, 2007) é um *game engine* didático para jogos eletrônicos em Java utilizando-se da biblioteca Java3d, desenvolvido e mantido pelo Interlab/USP (Laboratório de Tecnologias Interativas da Escola Politécnica da Universidade de São Paulo), disponível como *software* livre através da GPL.

O principal objetivo do enJine é servir como uma ferramenta de apoio ao ensino de computação, em particular computação gráfica (TORI; BERNARDES JR; NAKAMURA, 2006), sendo portanto didático, de aprendizado relativamente rápido e fácil. Sua arquitetura procura facilitar e incentivar as boas práticas de engenharia de *software*. Outro objetivo do enJine é servir como plataforma para o desenvolvimento de protótipos e testes de novas tecnologias, especialmente aplicadas em jogos eletrônicos.

O enJine fornece um conjunto de serviços para o desenvolvimento de jogos, bem como uma estrutura de classes que auxilia em sua organização. Suas principais características são:

- oferece um conjunto de classes Java que permitem a representação de diferentes entidades do jogo tais como personagens, jogadores e elementos de cenário;
- oferece uma infra-estrutura básica para a implementação de diferentes jogos, a qual permite interações complexas entre entidades do jogo através de um sistema de troca de mensagens;

- possui um sistema de detecção de colisões entre entidades do ambiente virtual, baseado em múltiplos estágios;
- implementa a renderização de cenas tridimensionais através da API Java 3D, de forma a utilizar grafos de cena e os recursos de placas aceleradoras gráficas;
- implementa a renderização de modelos animados através da técnica “*skin-and-bones*”, carregados do formato de arquivo .X;
- oferece recursos básicos para reprodução de áudio;
- possui camada para tratamento de dispositivos de entradas.

A arquitetura do enJine, conforme figura 12, é dividida em três camadas de abstração. A camada mais baixa representa as bibliotecas externas utilizadas para a comunicação com o sistema operacional e seus dispositivos, além de alguns outros serviços como a renderização tridimensional da API Java3D. A camada intermediária contém os principais módulos do enJine, responsáveis por implementar os serviços de jogos, como efeitos sonoros, detecção de colisão etc. A camada de mais alto nível implementa o *framework* do enJine, que é um conjunto de classes que disponibiliza uma solução mais completa e fácil de usar para o desenvolvimento de jogos.

Figura 12 - As três camadas da arquitetura do enJine, extraída de (BERNARDES JR; NAKAMURA; TORI, 2007).

A principal camada desta arquitetura, é a intermediária. Esta camada é composta por um conjunto de pacotes, apresentados na figura 13.

Figura 13 - Os principais pacotes do enJine, extraída de (BERNARDES JR; NAKAMURA; TORI, 2007).

No contexto deste trabalho o principal pacote é o `Core`, pacote que possui as classes que representam os objetos do jogo e a classe `GameState`.

O `GameState`, é utilizado, no enJine, para representar diferentes modos de operações dos jogos, como por exemplo telas de entrada, níveis ou fases dos jogos e telas de “*game over*”, e, também, para representar modos de operações internos no jogo, como um inventário em um jogo de RPG. Durante a execução de um jogo, sempre há um `GameState`, ativo, representando o estado no qual o jogo está operando naquele momento (BERNARDES JR.; NAKAMURA; TORI, 2007). Um jogo simples pode ser implementado fazendo uso de somente um `GameState`.

Outro pacote importante no enJine é o `input`, que define a camada de entrada do enJine, onde novos dispositivos de entrada podem ser acoplados, como no sistema de Rastreamento e Registro. Portanto, o comportamento da camada de entrada do enJine deve ser compreendido, a arquitetura desta camada pode ser vista na figura 14.

Figura 14 – Arquitetura da camada de entrada do enJine, adaptado de (BERNARDES JR; NAKAMURA; TORI, 2007).

Os diferentes dispositivos de entrada são definidos pela implementação concreta da classe `InputDevice`, que possui uma coleção de objetos `InputSensor`, que determinam suas capacidades. Cada `InputSensor` possui um identificador numérico, uma descrição e um valor em ponto flutuante, que pode ser usado também com valores discretos, funcionando como um botão ou uma tecla. Os valores destes sensores são atualizados pelo `InputDevice`.

O `InputManager` é responsável por manter o registro dos dispositivos de entrada e realizar a ligação entre cada `InputSensor` com uma `InputAction`. `InputActions` são objetos que representam comandos com objetivos específicos dentro do jogo, eles podem ser comandos do usuário como “pular” ou comandos disparados pelo sistema como “atualize a posição x”, no caso de um objeto rastreado.

4.2 ENJINE + ARTOOLKIT

Tsuda et al. (2007) apresentam uma integração da `JARToolKit` (GEIGER et al., 2002) com o enJine, tendo como objetivo facilitar a criação de jogos com AR de baixo custo, aumentando ainda mais o potencial do enJine como ferramenta didática e uma plataforma de testes.

A `JARToolKit` é uma biblioteca de classes implementadas em Java, que, através do JNI (*Java Native interface*) acessa as funções da biblioteca `ARToolKit`, originalmente desenvolvida em C/C++. A `ARToolKit` (ARTOOLKIT, 2007) é uma biblioteca que

fornece funções que facilitam o desenvolvimento de aplicações com AR. Baseada em Visão Computacional, ela oferece uma solução completa para o rastreamento de marcadores fiduciais, calculando a posição e orientação real da câmera em relação a esses marcadores, permitindo que objetos virtuais sejam renderizados sobre o ponto de visto correto, como se fizessem parte do mundo real.

A integração com a JARToolkit permite ao enJine a capacidade de reconhecer a posição e orientação real dos marcadores fiduciais e da câmera, além de reproduzir o vídeo obtido desta câmera, características que permitem o desenvolvimento de jogos com AR. Esta integração possui a arquitetura mostrada na figura 15, e discutida adiante:

Figura 15 - Arquitetura da integração enJine + jARToolKit, adaptada de (TSUDA et al., 2007).

A solução implementada para realizar a integração entre as classes do enJine e da JARToolkit foi o desenvolvimento de um componente de *middleware*, que possui as seguintes classes:

- `ARInstance`, classe que inicializa e acessa as funções da classe `jARToolKit3D`;
- `ARInput`, classe que representa um dispositivo de entrada, obtido através da análise das imagens capturadas pela câmera;
- `ARTimerTask`: estende a classe Java `TimerTask`, é responsável por obter a matriz de transformação do nó do grafo de cena que reconhece o marcador

na imagem e o transfere para a classe `ARInput`, em intervalos definidos pelo desenvolvedor do jogo;

- `ARDisplay`, estende a classe do `enJine GameDisplay`, permite a reprodução do vídeo capturado pela câmera e a adição do grafo de cena que representa os dispositivos de AR;
- `ARGame`, classe criada dentro da camada de *framework* do `enJine`, herança da classe `Game`, facilita a criação de jogos com AR, pois possui os atributos e métodos necessários para instanciar uma interface entre o `enJine` e a `JARToolKit` corretamente.

4.3 STUDIERSTUBE

Studierstube (STUDIERSTUBE, 2007) é um sistema criado com o objetivo de possibilitar a visualização de dados científicos tridimensionais por múltiplos observadores dentro de uma sala. Estes dados são imagens tridimensionais misturadas no ambiente real desta sala. Cada usuário deve usar um HMD estereoscópico, rastreado magneticamente, permitindo que ele se locomova livremente nesta sala, visualizando o conteúdo aumentado do ponto de vista desejado. Como forma de interação com o conteúdo virtual pode ser usada uma ferramenta denominada *Personal Interaction Panel* (PIP). Este dispositivo é um conjunto, composto de uma caneta e um *pad*, também rastreados magneticamente, que permite a manipulação de conteúdos tridimensionais ou formas de interação específicas da aplicação. A figura 16 mostra este dispositivo em uso, na imagem da direita pode ser visto o conteúdo aumentado.

Figura 16 - O dispositivo PIP em uso (STUDIERSTUBE, 2007).

A arquitetura de *software* para este sistema é baseada na estrutura cliente-servidor. Um servidor armazena todos os tipos de dados, incluindo registro, objetos móveis e dados de exibição ou da aplicação. Os usuários conectam-se ao Servidor de Ambiente através da rede utilizando o *software* cliente.

Os dados de rastreamento são gerenciados por um Servidor de Rastreamento. Esse servidor está configurado em uma máquina dedicada, para a obtenção de uma melhor performance. O sistema contém diversos pontos rastreados, e todos esses dados são propagados para os clientes conectados como um pacote, para garantir a consistência dos dados da cena. Um diagrama da arquitetura desse sistema pode ser visto na figura 17.

Figura 17 – A arquitetura do *Studierstube*, adaptada de (STUDIERSTUBE, 2007).

4.4 DWARF

O DWARF (*Distributed Wearable Augmented Reality Framework*) (MACWILLIAMS et al. 2003) é um *framework* baseado em serviços distribuídos (rastreamento de posições, renderização tridimensional, entrada e saída e modelagem de tarefas). Cada serviço é um programa que pode ser executado em componentes de *hardware* separados, com seu próprio processador, memória, dispositivos de entrada e saída e

comunicação em rede. Os diferentes serviços se conectam dinamicamente com outros serviços explicitando suas necessidades e habilidades através da rede. *Middlewares*, também distribuídos, são responsáveis por controlar as comunicações entre estes serviços. Esse *framework* é utilizado para desenvolver aplicações de AR completas, projetadas para serem flexíveis e garantir aos usuários um alto grau de liberdade. Essas características são atingidas devido ao fato desses serviços poderem ser distribuídos em várias máquinas independentes, que podem ser adicionados ou removidos do ambiente sem causar um grande impacto no sistema.

4.5 KOBITO

Kobito (AOKI et al., 2005) é um projeto que envolve AR e Inteligência Artificial, com o objetivo de demonstrar o potencial dessas tecnologias. Neste projeto, uma lata de chá, objeto real, é manipulada por criaturas virtuais, chamadas Kobito, que podem ser vistas por uma técnica de *video see-through* baseada em monitor chamada “*Kobito Window*”. Esta aplicação pode ser vista na figura 4.

Um dos principais objetivos desse projeto é explorar as possibilidades de interação entre os objetos reais e virtuais, de um modo que ambos possam se influenciar.

Uma das principais preocupações do projeto, para que os usuários possam ter a sensação de que os elementos virtuais estão “presentes” no mundo real, é a eliminação de marcadores, HMDs ou sensores acoplados em seu corpo, e também as formas que o usuário pode interagir com os elementos virtuais, que no caso é somente através da lata de chá.

O conceito e a arquitetura deste projeto podem ser vistos na figura 18. Para possibilitar essas interações a lata de chá é rastreada por uma câmera, que indica a sua posição na mesa, possibilitando que as criaturas virtuais “saibam” onde ela se encontra. Quando os elementos virtuais manipulam a lata de chá, esta é movida por um sistema (SPIDAR), que controla um eletroímã, debaixo da mesa. Na lata de chá está outro imã que segue os movimentos do primeiro. Todas as interações, entre os elementos virtuais e a lata de chá, a qual possui uma representação virtual, são calculadas por um sistema de simulação física.

Figura 18 - Arquitetura e conceito da aplicação Kobito, adaptada de (AOKI et al., 2005).

4.6 AUGMENTED COLISEUM

Augmented Coliseum (SUGIMOTO et al., 2005) é um jogo onde robôs reais competem em um ambiente virtual projetado, são incrementados com alguns efeitos como raios ou mísseis que os robôs podem disparar e explosões quando um robô é destruído. Esse ambiente também explora as interações possíveis entre os elementos reais e virtuais. Uma imagem dessa aplicação é mostrada na figura 19, onde podem ser vistos dois robôs, inseridos em um ambiente virtual, interagindo.

Figura 19 – Robôs em um ambiente virtual de *Augmented Coliseum*, extraída de(SUGIMOTO et al., 2005).

Essa aplicação utiliza uma tecnologia denominada DBC (*Display-Based Computing*), que faz uso de dispositivos de exibição para enviar informações ópticas. Nessa aplicação, essas informações são utilizadas para misturar a computação gráfica com robótica. Os robôs possuem um dispositivo com sensores de luminosidade que segue os movimentos de uma imagem de acordo com padrões estabelecidos. Esse sistema elimina a necessidade de um rastreamento baseado em captura de vídeo por câmeras.

4.7 CONSIDERAÇÕES FINAIS

Os trabalhos apresentados neste capítulo partilham diversas características com a Robot ARena, tais como a interação entre elementos virtuais e reais, rastreamento por visão computacional e foco em jogos. Como forma de simplificação e melhor visualização dessas características a tabela 2 faz uma síntese comparativa desses trabalhos.

Tabela 2 – Relação de Características por Projeto.

Característica / Projeto	enJine	EnJine + jARToolKit	Studierstube	DWARF	Kobito	Augmented Coliseum	Robot ARena
Infra-estrutura	sim	sim	sim	sim	não	não	sim
Interação Virtual – Real	não	não	não	não	sim	sim	sim
Rastreamento – Visão	não	sim	sim	sim	sim	não	sim
Rastreamento – Outros	não	não	sim	sim	não	sim	não
Foco em Jogos	sim	sim	não	sim	não	sim	sim
SAR	não	não	não	sim	não	sim	sim
Robôs	não	não	não	não	não	sim	sim
Interface Horizontal	não	não	não	sim	não	sim	sim

Essas características compartilhadas são resultantes da utilização e inspiração desses projetos sobre a infra-estrutura desenvolvida neste trabalho.

O enJine tem sido a tecnologia básica utilizada em diversos projetos do laboratório onde este trabalho foi desenvolvido. Desta forma, e principalmente por apresentar uma boa estabilidade e flexibilidade, o enJine foi adotado como estrutura fundamental para o desenvolvimento do *software* da infra-estrutura apresentada neste trabalho.

A integração enJine+jARToolKit inspirou este trabalho não somente na aplicação de AR no enJine, mas também na utilização da camada de entrada deste *game engine*.

Os projetos *Studierstube* e *DWARF* são exemplos de infra-estruturas que permitem diversas formas de visualização de AR e também permitem que diversos dispositivos de entrada sejam acoplados ao sistema, através de uma arquitetura flexível.

As aplicações *Kobito* e *Augmented Coliseum* demonstram uma área pouco explorada na AR, a influência dos objetos virtuais nos objetos reais. No projeto *Augmented Coliseum* esta interação é realizada através do uso de robôs telecomandados, assim como na *Robot ARena*.

5 DESENVOLVIMENTO DA ROBOT ARENA

Neste capítulo é descrito o projeto de desenvolvimento da Robot ARena. Sua visão geral, arquitetura e implementação são apresentadas, e, a partir de uma prova de conceito, são discutidos os resultados do projeto.

Como metodologia de desenvolvimento de *software* foi utilizado o Processo Unificado de Desenvolvimento de Software (UP, *Unified Process*) (JACOBSON; BOOCH; RUMBAUGH, 1999), que tem como objetivo promover o desenvolvimento de *software* progressivo, através de fases iterativas com etapas incrementais. Como em Bernardes Jr. (2004) o UP foi adaptado conforme as necessidades, abrangência e porte do projeto aqui descrito.

5.1 VISÃO GERAL

A Robot ARena é uma infra-estrutura, de *hardware* e *software*, destinada ao desenvolvimento de jogos e aplicações interativas que se utilizem de uma interface baseada em superfície, robôs telecomandados e projeções de realidade aumentada espacial (SAR). Esta infra-estrutura tem como base o enJine (ENJINE, 2007), motor de jogo que disponibiliza um *framework* para a criação de um mundo virtual e todos os recursos de *software* necessários para o desenvolvimento de jogos e ambientes interativos.

O robô, que dá nome à infra-estrutura, utilizado na Robot ARena tem o objetivo de ser o elemento de interação que pode ser afetado por elementos virtuais presentes no ambiente aumentado. Para isso o robô deve poder ser controlado remotamente. Como exemplo de implementação foi montado um robô utilizando o Lego Mindstorms (LEGO, 2007), e desenvolvido um módulo de controle e comunicação para o mesmo. Qualquer outro dispositivo móvel tele-controlado pode substituir a implementação do robô aqui descrita.

Funcionando como uma interface horizontal, a Robot ARena é montada sobre uma mesa, a qual permite a exibição de conteúdos virtuais tanto por projeção frontal

como por retro-projeção, utilizando o conceito de Dispositivos Espaciais Baseados em Projeção. A essa estrutura podem ser adicionadas câmeras com visão superior ou inferior da superfície, possibilitando ao aplicativo o rastreamento de elementos posicionados sobre a mesa. Com esta flexibilidade de configuração, a Robot ARena permite diversas formas de visualização baseadas em SAR e diversas possibilidades de interações, como reconhecimento de gestos e rastreamento de objetos, com uma câmera superior, ou a implementação de uma superfície multi-toques, através do reconhecimento de sombras ou a técnica empregada em (HAN, 2005), com uma câmera posicionada na parte inferior da infra-estrutura.

Para atingir esta flexibilidade foi utilizada a camada de entrada fornecida pelo enJine. Uma das especializações dos dispositivos de entrada é o componente de rastreamento de objetos, que pode ser utilizado para atualizar a posição das representações dos objetos reais no ambiente virtual. Esta implementação permite o rastreamento de um marcador, que pode ser montado junto ao robô ou utilizado como um dispositivo de interface tangível, na Robot ARena.

5.2 ARQUITETURA

A Robot ARena foi implementada na estrutura de um jogo, fornecida na figura 9, como uma camada de abstração logo acima do enJine, mas ainda a este pertencendo, criando novas possibilidades de código e de visualização do jogo. A figura 20 mostra a Robot ARena inserida na estrutura do jogo.

Figura 20 - Estrutura de um jogo com a camada da Robot ARena, baseada em (LEWIS; JACOBSON, 2002).

Como uma visão mais específica da camada da Robot ARena, a figura 21 mostra a arquitetura dessa infra-estrutura.

Figura 21 – Arquitetura da Robot ARena.

A parte física da Robot ARena pode ser composta pela mesa, ou superfície de projeção, por um espelho à 45°, por uma ou mais câmeras, um ou mais projetores e pelo robô ou quaisquer objetos reais rastreados e/ou telecomandados, além do computador.

O sistema de *software* dessa infra-estrutura foi dividido em quatro subsistemas, sendo que alguns desses necessitam acessar informações de elementos da parte física, que, como forma de simplificação, foram considerados “caixas pretas” que enviam e recebem sinais. Esses subsistemas são descritos abaixo:

- **Visualização** - O subsistema de visualização fornece uma classe que determina os parâmetros da câmera dentro do ambiente virtual, para que a projeção seja corretamente calibrada na superfície de projeção desejada. Esta etapa de calibragem pode ser resolvida automaticamente por visão computacional, ou manualmente pelo usuário. Este subsistema foi

implementado dentro de um `GameState`, permitindo que este seja acoplado em qualquer jogo desenvolvido, ou executado separadamente.

- *Rastreamento e Registro* - Este subsistema se traduz em um componente que disponibiliza uma interface que possui métodos para realizar o registro entre os ambientes real e virtual, e o rastreamento de objetos reais inseridos na aplicação. Para futuras implementações este subsistema serve como padrão, exemplificando o uso da camada de entrada do enJine, permitindo que outras formas de interação sejam facilmente integradas à Robot ARena.
- *Elementos Reais* - Um conjunto de classes que, por herança da classe `GameObject` do enJine, especificam as características dos objetos do jogo que possuem uma representação física. A posição real destes elementos físicos é determinada pelo sistema de Rastreamento e Registro.
- *Comunicação e Controle do Robô* - Um conjunto de classes que define um robô e fornece a interface para a comunicação e controle do robô real, permitindo que o jogador envie comandos de movimento para o robô, assim como dos elementos virtuais, que também podem interagir com o robô. Este componente envia as informações de acordo com a atualização fornecida pela classe `RealUpdater`. Outros componentes podem ser acoplados de acordo com a tecnologia empregada no desenvolvimento do robô.

Esses subsistemas podem ser utilizados de acordo com a aplicação desejada, e outros subsistemas podem ser adicionados, garantindo-se a flexibilidade da arquitetura. Um exemplo é uma aplicação que utilize reconhecimento de gestos como forma de interação em um jogo. O desenvolvedor utilizará o subsistema de visualização para calibrar a projeção de ambiente virtual e acoplará um novo subsistema com um dispositivo de entrada, similar ao Rastreamento e Registro, que envia as informações dos gestos reconhecidos.

5.3 IMPLEMENTAÇÃO

Nesta seção são descritas as implementações dos elementos físicos e do sistema de *software* que compõem o protótipo da Robot ARena bem como as ferramentas e tecnologias que foram utilizadas. Como a metodologia deste projeto se baseia numa abordagem incremental e iterativa, serão relatadas, onde pertinentes, as etapas, soluções e ferramentas anteriores à implementação final do protótipo.

5.3.1 Tecnologias Utilizadas

Além do enJine, previamente descrito, que foi adotado como base fundamental para o desenvolvimento do *software* da infra-estrutura apresentada neste trabalho, abaixo são descritas outras tecnologias e ferramentas utilizadas.

5.3.1.1 *OpenCV*

A OpenCV (*Open Source Computer Vision Library*) (OPENCV, 2007) é uma biblioteca gratuita desenvolvida em C/C++ que implementa diversas funções e algoritmos de Processamento de Imagens e Visão Computacional, com um foco em aplicações em tempo-real, o que a torna interessante para seu uso em AR. Dois dos principais requisitos de AR é o registro e rastreamento e uma das formas de obtê-los é através de Visão Computacional, o que torna a OpenCV uma ótima escolha para estas aplicações, por ser flexível e robusta.

5.3.1.2 *Lego Mindstorms*

O Lego Mindstorms (LEGO, 2007) ou Lego Robotics Invention System (RIS), é um conjunto de blocos, motores, sensores e outros componentes, como engrenagens e

eixos, criado e vendido pelo grupo Lego. Este conjunto pode ser utilizado para montar diversos tipos de robôs e outros sistemas automatizados ou interativos.

Para o desenvolvimento deste projeto foram utilizadas as duas versões deste kit. No primeiro protótipo foi utilizada a versão RCX 2.0, nome do principal componente do Lego Mindstorms, que controla os sistemas de entrada e saída (sensores e motores). Este componente possui um microcontrolador Renesas H8/300 que interpreta os programas, carregados em sua memória RAM através de uma interface que utiliza infravermelho para receber e enviar dados.

A segunda versão utilizada foi, a mais atual, Mindstorms NXT, que possui um microprocessador Atmel 32-bit ARM, e uma interface de comunicação por USB ou, sem fio, através de Bluetooth.

5.3.1.3 ICommand

ICommand (LEJOS, 2007) é uma biblioteca em Java para o controle do Lego NXT através de uma conexão Bluetooth. Para a comunicação Bluetooth é utilizada a biblioteca BlueCove (BLUECOVE, 2007), que implementa a JSR-82, junto com a RXTX (RXTX, 2007) para a comunicação através de portas seriais emuladas pelo Bluetooth.

5.3.1.4 JavaComm

A Java Communications API (JavaComm) (JAVA, 2007) é uma biblioteca Java que auxilia no desenvolvimento de aplicações de comunicação independente de plataforma, para tecnologias como: sistemas embarcados, fax, modem, terminais de exibição e, como neste trabalho, robótica. Esta biblioteca permite que as aplicações acessem o hardware RS-232 (porta serial) e IEEE-1284 (porta paralela).

A JavaComm foi utilizada para estabelecer uma comunicação com a primeira versão do robô, que era construído com o Lego Mindstorms RCX 2.0.

5.3.1.5 Câmera

Duas câmeras foram utilizadas durante o desenvolvimento do projeto. A primeira versão foi uma Creative NX Pro, que possui um sensor CMOS VGA com resolução de 640x480, com interface de comunicação através de USB 1.1, um campo de visão de 35°, e uma taxa de captura de 15 quadros por segundo.

A segunda câmera utilizada foi uma C3 LT-359A, com um campo de visão de 55°, que permite o enquadramento total da superfície da Robot ARena, um sensor CMOS VGA com resolução de 640x480, e comunicação através de USB 2.0, permitindo uma taxa de captura de até 30 quadros por segundo.

Algumas outras câmeras foram testadas como a LG WebPro2 e a Logitech QuickCam 4000 e 5000, com o objetivo de comparar a sensibilidade destas câmeras à iluminação infravermelha, porém todas *webcams* testadas possuíam filtros bloqueadores de infravermelho, de difícil remoção, podendo danificar as lentes das câmeras caso sejam removidos. Devido a este fator não foi implementado o dispositivo rastreável infravermelho no protótipo final da Robot ARena.

5.3.1.6 Projetor

O projetor utilizado é um Hitachi CP-X255, com as seguintes características: 3 LCDs, resolução XGA de 1024x768, 2000 ANSI lumens e contraste de 400:1. Uma importante característica deste projetor é a distância do arremesso de sua projeção, que permite, utilizando o zoom, uma área projetada de 120cm x 90cm a uma distância de 180cm, que facilita a posicionamento do projetor na Robot ARena, tanto para a configuração com projeção frontal como para retro-projeção.

5.3.2 Parte Física

Nesta seção é discutida a evolução e a implementação da parte física da Robot ARena.

5.3.2.1 A Arena

Inicialmente, a versão para prova de conceito da infra-estrutura, possuía uma configuração fixa, com uma câmera, um projetor e um espelho, montados sobre uma mesa, permitindo apenas o uso de projeções frontais. Uma representação desta configuração inicial pode ser vista na figura 22.

Figura 22 – A primeira configuração da Robot ARena.

A versão final da Arena foi desenvolvida tendo como objetivo oferecer maior flexibilidade, permitindo diversos tipos de configurações de projetores e câmeras. A figura 23, mostra a representação da Arena.

Figura 23 – Versão final da Robot ARena.

A Arena construída para testes do protótipo do sistema possui 140cm de largura por 110cm de comprimento e 100cm de altura. Com o seu tampo vazado, faz uso de um acrílico cristal coberto por uma tela que permite retro-projeções, possui uma área útil de projeção com: 120cm x 90cm, mantendo a proporção de 4:3. Na parte inferior da Arena está montado um espelho, 90cm x 130cm, à 45°, que permite o direcionamento das imagens para retro-projeção, e a visualização da parte inferior da superfície de projeção por uma câmera.

Para a implementação da prova de conceito e protótipos de jogos foi utilizada a configuração com retro-projeção e câmera superior, primando pela praticidade e simplicidade de montagem, como afirma (WILSON, 2005).

5.3.2.2 O Robô

A primeira versão do robô foi montada utilizando o Mindstorms RCX 2.0 da Lego. O robô, possuía uma montagem com esteiras e utiliza um sistema diferencial de direção, que possibilitava uma rotação sobre seu próprio eixo, sem causar nenhuma alteração na sua posição, característica importante para a simplificação do cálculo

de detecção de colisões com os objetos virtuais. O primeiro robô montado pode ser visto na figura 24.

Figura 24 – Primeira versão do robô.

Como esta versão do robô não possuía uma forma de comunicação sem fio, a comunicação entre o computador e o robô era realizada através de um módulo de transmissão de radiofrequência, no computador, e um módulo de recepção, no robô, cada módulo possui um transceptor de 2.4Ghz, micro-controlado (CALIFE et al., 2006 e 2007). A figura 25 mostra um diagrama do sistema de comunicação, que envia os dados do usuário para os sensores de entrada do robô, que ativam movimentos programados.

Figura 25. Módulo de comunicação da primeira versão do robô.

A segunda versão do robô, montado com o Lego NXT, permitiu a comunicação através de uma conexão Bluetooth e com o uso do ICommand possibilitou um maior controle das características e funções do robô, como controle de velocidade de cada motor do robô, a utilização de efeitos sonoros e outras funcionalidades do NXT. Isto significa que, os movimentos e respostas do robô não precisam ser previamente

programados e apenas responder às entradas dos sensores. Em linhas gerais a montagem da versão final do robô segue a mesma configuração da versão anterior, com esteiras e sistema diferencial de direção. A figura 26 mostra o robô montado com o NXT.

Figura 26 – Versão final do robô, montado com o Lego NXT.

5.3.2.3 Dispositivos de Rastreamento

O primeiro protótipo da Robot ARena possuía a característica de rastrear dois dispositivos diferentes. Um deles era montado com blocos Lego, formando um marcador colorido, o outro era um dispositivo com dois LEDs infravermelhos.

O marcador colorido foi criado especialmente para o rastreamento do robô. Seus dois blocos Lego, com diferentes cores, desenham um “T”, que pode ser montado em cima do robô, este marcador é mostrado na figura 27.

Figura 27 – Marcador colorido, montado com blocos Lego.

O dispositivo de rastreamento infravermelho é uma pequena caixa com dois LEDs, que pode ser facilmente manipulada por usuários ou associada com algum outro objeto que se deseje rastrear. Seus dois LEDs têm uma função similar ao uso das cores no marcador Lego. Para diferenciar os dois LEDs um circuito digital regula a intensidade de ambos, fazendo com que um deles possua um brilho maior que o do outro, permitindo assim, a identificação de cada um. Uma representação deste dispositivo pode ser vista na figura 28.

Figura 28 – Dispositivo de rastreamento infravermelho.

Na versão final do protótipo da Robot ARena foi utilizado um novo marcador Lego, uma vez que o dispositivo infravermelho foi descartado devido à variação de sensibilidade das câmeras e bloqueadores de IR presentes em *webcams*. Este marcador possui duas cores, e sua montagem foi baseada no algoritmo de segmentação apresentado em (VAN DEN BERGH; LALIOTI, 1999), que segmenta apenas um canal de cor da imagem, neste caso o canal amarelo. Este marcador pode ser visto na figura 29.

Figura 29 – Marcador Lego amarelo.

5.3.3 Software

Nesta seção será discutida a evolução e a implementação do *software* da Robot ARena.

5.3.3.1 Visualização

O subsistema de Visualização foi formado por uma especialização da classe `GameState` do `enJine`, `CalibrationState`, que tem como um de seus atributos um objeto da classe `Calibration`, implementada em C++ com o auxílio da OpenCV. Para a integração destas classes foi utilizado o JNI (*Java Native Interface*), que define uma interface para criar uma DLL em C++, permitindo a chamada de seus métodos dentro de uma classe Java.

Neste subsistema o `CalibrationState` representa o modo de calibragem da visualização, projeção, de um jogo, que pode ser um estado antes do início do jogo ou, caso a configuração da Arena seja fixa, uma aplicação que é executada uma única vez, permitindo gravar seus parâmetros, que só precisam ser atualizados caso haja alguma modificação na configuração da Arena. O diagrama de classes deste subsistema é mostrado na figura 30.

Figura 30 – Diagrama de classes do subsistema de Visualização.

A classe `CalibrationState` possui uma instância de `Camera`, que representa a câmera virtual dentro do jogo, no `enJine`. Portanto o campo de visão e o posicionamento desta câmera representam a imagem que será projetada na Arena. Assim, é preciso calibrar esta câmera, para garantir o alinhamento da imagem gerada, como a superfície de projeção.

`CalibrationState` basicamente exibe uma imagem padrão, dois quadrados brancos sendo um deles maior, que devem ser alinhados à superfície de projeção. Este alinhamento pode ocorrer manualmente, através de comandos do teclado, que permite que a câmera virtual seja rotacionada e transladada, até que o padrão seja alinhado e centralizado na Arena. Este processo de alinhamento também pode ser feito automaticamente através da classe `Calibration`, que realiza a calibragem a utilizando técnicas de visão computacional, segmentando estes quadrados projetados e definindo sua posição atual para calcular a posição correta da projeção. Em ambos os casos os parâmetros gerados podem ser salvos em um arquivo XML, e carregados posteriormente, não necessitando uma nova etapa de calibragem.

A calibragem é realizada através de etapas, que devem seguir esta ordem, mesmo quando realizada manualmente:

- Determinar e corrigir a orientação da câmera virtual, `getOrientation()`;
- Determinar um fator de escala de *pixels* para unidades genéricas 3D, `getPixelto3d()`;
- Corrigir a posição em X e Y da câmera, o eixo Z é fixo pois a câmera mantém uma visão superior do jogo (assim como a Arena), `getXTrans()` e `getYTrans()`;
- Determinar o tamanho do robô real, caso necessário, `getRobotSize()`;
- Salvar os parâmetros encontrados, `saveParams()`.

5.3.3.2 Rastreamento e Registro

O subsistema de Rastreamento e Registro foi dividido em duas classes, a classe `ObjectTracker`, implementada em C++ utilizando o OpenCV, que determina a

postura, posição e a orientação do objeto rastreado, e a classe `Tracker`, implementada em Java utilizando o `enJine`, que define o dispositivo de entrada que, através de seus sensores, atualiza o estado lógico da representação virtual do objeto rastreado. Integradas utilizando o `JNI`, assim como no subsistema de Visualização.

Para a implementação de subsistema foi utilizada a camada de entrada do `enJine`, através de um dispositivo de entrada, assim como este subsistema outros dispositivos de entrada podem ser criados, permitindo que outras formas de interação sejam facilmente integradas à `Robot ARena`.

Utilizando esta camada, o dispositivo de entrada `Tracker`, implementação concreta da classe abstrata `InputDevice`, é responsável pela atualização da posição lógica do objeto real rastreado. O diagrama de classes do subsistema de Rastreamento e Registro é mostrado na figura 31.

Figura 31 – Diagrama de classes do subsistema de Rastreamento e Registro.

A utilização da classe `ObjectTracker` foi dividida em duas etapas, a inicialização e o rastreamento. Na etapa de inicialização, `init()`, a câmera é iniciada; seus parâmetros intrínsecos são determinados, com o objetivo de remover possíveis distorções das lentes; é calculada uma matriz de homografia, para que a perspectiva da câmera seja corrigida de forma que os quatro cantos da superfície de projeção

sejam alinhados com as extremidades do quadro capturado, permitindo que a câmera seja posicionada livremente, com a única exigência de sempre enquadrar os quatro cantos da superfície de projeção da ARena; e, a relação pixel por milímetros é calculada, para determinar a posição real do objeto rastreado na superfície da ARena.

Para que a matriz de homografia seja calculada é necessário encontrar os quatro cantos da superfície de projeção. Com o objetivo de facilitar esta procura quatro adesivos vermelhos foram posicionados nos cantos da superfície e através de uma limiarização no canal vermelho do quadro capturado os adesivos são segmentados na imagem, representando os pontos de origem para o cálculo da matriz (os pontos destino são as extremidades dos quadros capturado pela câmera, dependendo da sua resolução). Estes pontos também são utilizados para determinar a relação entre pixels e milímetros, uma vez que a distância real entre estes pontos é conhecida. A figura 32 (a) mostra a imagem original capturada pela câmera e em (b) a imagem após a correção das lentes e homografia.

Figura 32 – (a) imagem capturada pela câmera; (b) imagem após correções.

Partindo-se do princípio de que a câmera esteja fixa em uma posição, estes parâmetros não se alteram. Portanto a classe `ObjectTracker` fornece um método, `saveParams()`, que salva em um arquivo estes parâmetros, que pode ser passado para o construtor de `ObjectTracker`.

Na etapa de rastreamento a posição, em X e Y, e a orientação do objeto rastreado são determinadas utilizando o método `calcNewPosition()`, e depois estes dados são lidos através dos métodos `getX()`, `getY()`, `getOrientation()`. A abordagem utilizada para o rastreamento é baseada na segmentação de cores, por

isso a utilização de dispositivos rastreáveis montados com blocos Lego coloridos. Na técnica descrita por van den Bergh e Lalioti (1999) normalmente o canal azul de uma imagem RGB é segmentado, porém as peças Lego desta cor não apresentam uma pureza significativa neste canal, portanto foram utilizadas peças amarelas, que pode ter seu canal amarelo segmentado, uma vez que a imagem seja convertida para o padrão CMYK.

Após a segmentação das cores do marcador, são encontrados dois componentes conexos, e a partir dos centros destes componentes é determinada a posição, média entre estes centros, e a orientação, após determinar a origem e o destino pelo número de pixels que formam os componentes conexos (o menor é a origem) é calculado o ângulo deste vetor.

O objeto de `ObjectTracker` é instanciado dentro da classe `Tracker`, realiza o cálculo de rastreamento e a atualização dos valores de seus sensores (`xPos`, `yPos` e `orientation`) em uma *thread* separada.

5.3.3.3 Elementos Reais

O subsistema de Elementos Reais foi formado por classes que, por herança da classe `GameObject` ou `Updater`, presentes no pacote `Core` do `enJine`, especificam as características dos objetos do jogo que possuem uma representação física. O diagrama de classes deste subsistema pode ser visto na figura 33.

Figura 33 – Diagrama de classes do subsistema de Elementos Reais.

A classe `RealObject` é herança de `GameObject`, e basicamente implementa os atributos e métodos necessários para definir e manipular um objeto real, que pode ser rastreado ou não. Portanto, possui como atributos `position` e `rotation`, que determinam sua postura, e métodos para ler e atualizar estes atributos.

A classe `RealUpdater`, especializa a classe `Updater` do engine, e possui como atributo um dispositivo de entrada `Tracker`, e os `InputActions`, que durante a execução do método `updater()` são atualizados pelos `InputSensors` de `Tracker`. Os `InputActions`, por sua vez, atualizam a posição e orientação do `RealObject`. Outros dispositivos de entrada de rastreamento podem ser utilizados para atualizar a posição dos `RealObjects`, ao invés do `Tracker` desenvolvido neste trabalho.

5.3.3.4 Comunicação e Controle do Robô

O subsistema de Comunicação e Controle do Robô foi implementado com a classe `RobotController`, que compõe a classe `Robot`. A classe `Robot`, é uma especialização da classe `RealObject`, e possui como atributo um `RobotController`, e implementa os métodos para controlar os movimento do robô. O `RobotController` é uma class abstrata, e deve ser implementada de acordo com a tecnologia utilizada para a criação do robô.

Para o robô criado com o Lego NXT, esta classe foi implementada com a ajuda da biblioteca `ICommand`, que inicia e controla a comunicação através de uma conexão Bluetooth, fazendo uso das bibliotecas `BlueCove` e `RXTX`. A figura 34 mostra o diagrama de classes.

Figura 34 – Diagrama de classes do subsistema de Comunicação e Controle do Robô.

Neste caso a implementação concreta da classe `RobotController` possibilitou, através de classes do `ICommand`, os movimentos básicos do robô, acionando ou parando seus motores. Ao contrário da primeira versão do controlador do robô, baseado nas limitações do RCX, é possível controlar a velocidade de cada motor, o número de rotações, tocar clipes de sons, e exibir imagens de ícones na tela do NXT. Por uma limitação do sistema de comunicação do Lego NXT, só é possível comunicar com um robô.

Este subsistema permite que o jogador envie comandos de movimento para o robô, assim como os comandos dos elementos virtuais, que também podem interagir com o robô.

5.4 PROVA DE CONCEITO E TESTES

Como teste para a infra-estrutura, o protótipo de um jogo foi proposto, servindo de prova de conceito para explorar as capacidades da Robot ARena. Embora com regras simples, o protótipo oferece o ambiente necessário para a interação do robô com os elementos virtuais presentes no jogo.

A mecânica básica do jogo é movimentar o robô para que ele colete as baterias virtuais projetadas. Como obstáculos, existem barris que impedem a passagem do robô, caso o robô colida com um destes barris, ele não poderá avançar, sendo obrigado a recuar e dar a volta no obstáculo.

Os modelos virtuais do jogo foram criados com o auxílio de um *software* de modelagem 3d e são simples objetos poligonais com alguns materiais aplicados. Toda parte de interação entre estes elementos e o robô real foi realizada através do sistema de detecção de colisão fornecido pelo enJine.

A figura 35 mostra uma imagem do jogo em execução, onde o robô real está movimentando-se em direção ao obstáculo.

Figura 35 – Imagem da prova de conceito da Robot ARena.

A prova de conceito e os testes foram implementados utilizando um computador com as seguintes configurações: processador Intel Core 2 Quad Q6600 2.40GHz, com uma memória de 4GB de RAM e uma placa aceleradora gráfica Nvidia Geforce 7300 LE com 256MB de memória dedicada.

O sistema de rastreamento permitiu que a posição e a orientação de um objeto real fossem determinadas, com uma taxa de atualização que permite interações em tempo real. Ao se utilizar uma resolução de 640x480 *pixels* o algoritmo de rastreamento, executado isoladamente, atingiu uma taxa média de atualização de 10 quadros por segundo, ou seja um tempo de execução de 100ms. Do total deste tempo, 40ms foram gastos apenas com a correção do quadro obtido pela câmera, correção das lentes e homografia. A precisão obtida com esta resolução é de aproximadamente 4mm².

Com a utilização do algoritmo de segmentação por cor, baseado em técnicas descritas por van den Bergh e Laloti (1999) e a utilização de apenas uma cor para essa segmentação, houve uma sensível redução de *jittering*, presente quando

utilizado o marcador de duas cores (CALIFE; BERNARDES JR.; TORI, 2007) e também, uma redução na interferência da luminosidade ambiente, eliminando a necessidade do uso do marcador infravermelho, que se mostrou de difícil implementação, devido às diferenças de sensibilidade à luz infravermelha nas *webcams* testadas, sendo que estas possuem filtros bloqueadores de infravermelho, de difícil remoção, podendo danificar as lentes das câmeras caso sejam removidos.

Medidas de desempenho foram obtidas da prova de conceito como um todo, e, incluindo todo o processo de renderização 3D, a detecção de colisões, o sistema de comunicação com o robô mais o sistema de rastreamento, a taxa de atualização ficou estável em 60 quadros por segundo, estes limitados pela taxa de atualização do projetor utilizado (60Hz). Esta taxa não sofreu influência do sistema de rastreamento por este ter sido implementado em uma *thread* separada.

5.5 ANÁLISE DE RESULTADOS E CONSIDERAÇÕES FINAIS

Considerando os objetivos apresentados para a infra-estrutura, os resultados apresentados foram satisfatórios e promissores em todas as técnicas implementadas para esta configuração da Robot ARena.

A calibragem da projeção para a prova de conceito foi realizada automaticamente, facilitando o posicionamento do projetor pelo usuário, pois mesmo que a projeção não fique alinhada com a superfície, o processo de calibragem automático irá corrigi-la. Como foi adotada uma configuração fixa para a execução da prova de conceito, optou-se por implementar o `CalibrationState` como uma aplicação independente, um jogo, que era executado apenas uma vez, enquanto as configurações da Arena eram mantidas. Os parâmetros foram salvos e carregados através de um arquivo XML.

A OpenCV foi uma ferramenta importante, fornecendo uma camada de abstração para uma comunicação com a câmera, e possuindo funções diretas e rápidas de processamento de imagens e visão computacional, como as funções utilizadas para o cálculo da homografia.

O robô montado, utilizando o Lego Mindstorms NXT, junto com a biblioteca ICommand, possui uma flexibilidade e um controle maiores que a da versão anterior, RCX, e permitiu um maior nível de interação entre o usuário e o robô. Com uma conexão Bluetooth rápida, o tempo de resposta entre um comando enviado e o movimento do robô foi imperceptível, dificultando sua medição exata.

O enJine provou ser uma escolha adequada como núcleo da infra-estrutura, facilitando a criação do ambiente virtual, a integração entre os diferentes subsistemas e para o desenvolvimento das mecânicas do jogo.

Com base no sucesso da implementação desta configuração da Robot ARena, e dos resultados aqui demonstrados, outros dois protótipos de jogos foram desenvolvidos e são apresentados no capítulo 6.

6 PROTÓTIPOS DE JOGOS DESENVOLVIDOS

Este capítulo apresenta dois protótipos de jogos desenvolvidos para a Robot ARena, cada um com o objetivo de explorar e testar certas características desta infraestrutura.

6.1 FOOTBOT ARENA

O objetivo do desenvolvimento do protótipo do FootBot ARena foi explorar as interações entre os elementos reais e virtuais.

O jogo consiste de uma partida de futebol entre dois robôs, um real e o outro virtual, jogando com uma bola virtual, em um campo. Aquele robô que marcar três gols é o vencedor da partida. Embora existam muitas similaridades com as competições de futebol de robôs reais (ROBOCUP, 2007), este jogo não teve como objetivo simular ou duplicar as características destas competições.

Como incremento à jogabilidade da disputa entre os robôs, durante as partidas são disponibilizados itens que os robôs podem coletar, adquirindo assim algumas habilidades que os ajudam a atingir seus objetivos. Estes itens são:

Bomba: este item permite ao robô lançar uma bomba, que rebate por todo o campo até acertar um robô (pode ser inclusive o próprio robô que a lançou), quando um robô é atingido ele fica paralisado por 5 segundos.

Raio: permite ao robô que o capturar atirar um raio em direção ao robô oponente, caso um robô seja atingido três vezes ficará paralisado por 10 segundos.

Escudo: cria um escudo de proteção ao redor do robô, o protege contra raios e bombas por 20 segundos.

Outros elementos presentes no campo da partida são obstáculos que impedem a passagem de ambos robôs, e atrapalham o “chute” da bola ao gol adversário.

A figura 36 mostra o jogo FootBot ARena em execução na Robot ARena, na figura pode ser visto o robô real controlando a bola virtual e a representação de um robô virtual.

Figura 36 – Protótipo FootBot ARena em execução.

Assim como visto nos resultados da prova de conceito da Robot ARena, a interação entre os elementos reais e virtuais foi realizada com sucesso, criando a impressão que ambos, real e virtual, coexistem. No FootBot ARena, o que destaca ainda mais esta impressão é o fato de que o comportamento de ambos robôs, real e virtual, foi exatamente o mesmo, principalmente no que diz respeito à utilização e interação com os itens, obstáculos e com a bola presente no jogo. Porém, como o objetivo deste protótipo foi explorar ainda mais as interações entre esses elementos, foram realizados alguns testes para que os elementos virtuais pudessem interagir mais ativamente nos elementos reais, neste caso o robô Lego, como por exemplo este ser empurrado pelo robô virtual. Os resultados obtidos não foram satisfatórios. O fato principal que dificultou esse tipo de interação foi a configuração do robô Lego, que permite apenas movimentos para frente ou para trás e rotações em seu próprio eixo, impossibilitando que este se mova lateralmente, caso o robô virtual assim o empurrar. Este problema também ocorreu com o robô virtual, uma vez que este possui as mesmas características de movimentação do robô real. Caso o robô montado não possuía essas limitações, e possa mover-se em qualquer direção, seria necessário apenas implementar um tratamento de colisão, utilizando o enJine, que pudesse enviar os comandos adequados para o robô real.

6.2 TANSPLACE

O principal objetivo deste jogo foi explorar a implementação da TUI (*Tangible User Interface*) utilizando o dispositivo rastreável da Robot ARena, interagindo com o conteúdo projetado.

O jogo se passa no espaço, sendo que o jogador encontra-se dentro de uma nave que deve abater as naves inimigas. Para isso ele deve controlar uma mira até a nave inimiga e alinhá-la com uma segunda mira, que se movimenta junto com a nave. A figura 37 mostra a ambientação espacial do jogo, a mira que o jogador controla e a nave inimiga, com a segunda mira, que o jogador deve abater.

Figura 37 – Elementos do jogo TanSpace.

Naturalmente um dispositivo de TUI é capaz de determinar a posição desta mira, permitindo assim ao jogador posicionar livremente o dispositivo conforme a posição da nave inimiga e acompanhá-la. Porém, como o principal objetivo deste tipo de interface foi livrar os usuários do teclado e mouse, houve o problema de se enviar o comando para atirar na nave inimiga. Para solucioná-lo, criou-se para cada nave inimiga uma indicação, na forma de setas, que define a melhor posição para o tiro, e a própria mira que o jogador controla também possui setas, que devem ser alinhadas com estes indicadores da nave, figura 37. Quando a mira está posicionada em cima da nave inimiga, e ambas as setas estão alinhadas, a nave do jogador dispara um tiro causando a destruição da nave inimiga. Deste modo o jogo fez uso de ambas as características da interface tangível desenvolvida, a determinação da

posição e da orientação, e livrou os jogadores do teclado e do mouse, possibilitado uma interação muito mais simples e intuitiva.

A figura 38 mostra o jogo em execução na Robot ARena, onde pode ser vista a mira vermelha, que o jogador controla através do dispositivo tangível, e uma nave inimiga que possui uma indicação para o alinhamento de ambas as miras.

Figura 38 – Protótipo TanSpace em execução.

Durante os testes do TanSpace percebeu-se que devido ao formato do dispositivo rastreável, que neste atua como elemento tangível, ocorreu diversas vezes a obstrução do marcador pela mão do usuário, ocasionando erros no cálculo de sua posição e orientação. Esta obstrução ocorreu, na maioria das vezes, nos movimentos de rotação do marcador, movimento realizado pelos jogadores com o objetivo de alinhar as miras e destruir a nave inimiga.

6.3 ANÁLISE DOS RESULTADOS E CONSIDERAÇÕES FINAIS

Em ambos os protótipos de jogos algumas dificuldades foram encontradas. No FootBot ARena, uma possível solução para que seja possível um elemento virtual ativamente movimentar o robô real é a implementação de um sistema de detecção e retorno de colisões mais elaborado, permitindo-se que forças aplicadas em sua lateral resultem na sua rotação, ao invés de sua movimentação lateral.

Para o jogo TanSpace, uma possível solução para o problema da oclusão do marcador é a elaboração de uma base, sobre a qual o marcador colorido pode ser montado. Assim o jogador passa a manipular esta base, e não o próprio marcador, não ocasionando sua obstrução.

Embora utilize um robô real que pode ser influenciado por elementos virtuais, o jogo FootBot ARena poderia obter a mesma jogabilidade fazendo uso apenas de elementos virtuais. Mas a experiência vivenciada pelo jogador é muito mais rica ao permitir-lhe que modifique ou crie seu próprio robô e ao eliminar a visualização do jogo através de um monitor, passando para uma visualização mais ampla no próprio ambiente real.

O jogo TanSpace por sua vez, possui uma jogabilidade única, obtida através da aplicação de dois tipos de interfaces discutidas neste trabalho, as Interfaces Horizontais e Tangíveis, permitindo que o jogador interagisse com o jogo diretamente sobre a mesa, e superfície de visualização, remetendo às interações em jogos de mesa tradicionais.

Retornando à Tabela 1, apresentada no capítulo 3, pode-se conferir que diversos aspectos presentes nos jogos tradicionais e jogos eletrônicos foram combinados em ambos os jogos aqui apresentados, como a resolução de regras e cálculos, apresentação de um ambiente virtual e, ainda sim, mantendo um aspecto mais social e físico no jogo.

O aspecto social poderia, ainda, ser mais explorado no jogo TanSpace, evoluindo o subsistema de Rastreamento e Registro para que este permita dois dispositivos rastreáveis simultâneos, possibilitando modos de jogos cooperativos ou competitivos no jogo.

O principal resultado obtido com o desenvolvimento desses protótipos é que a infraestrutura Robot ARena disponibiliza os recursos necessários para que diferentes tipos de jogos, que explorem novas tecnologias interativas baseadas em realidade aumentada espacial, possam ser desenvolvidos e testados.

7 CONCLUSÃO

A principal contribuição deste trabalho é a Robot ARena, uma infra-estrutura que auxilia no desenvolvimento e pesquisa de jogos e aplicações interativas, em uma interface horizontal, utilizando-se de realidade aumentada espacial para a visualização do conteúdo virtual, não fazendo uso, assim, de quaisquer dispositivos acoplados ao usuário.

Outra importante contribuição é a base fornecida para a exploração do uso de objetos telecomandados, neste trabalho implementado com um robô, que permite a interação em ambas as direções entre elementos virtuais e elementos reais. Característica que reforça a sensação de que o ambiente virtual e o ambiente real coexistem.

A utilização do enJine como núcleo do software garantiu à Robot ARena a flexibilidade e a expansibilidade desta infra-estrutura, características que são marcantes neste *game engine*, e norteiam seu desenvolvimento. Por sua vez, a Robot ARena expande os aspectos didáticos presentes no enJine, possibilitando, por exemplo, o estudo de visão computacional e robótica, e, também, afirma sua vocação como ambiente de testes para tecnologias interativas emergentes.

Além da própria Robot ARena e o enJine poderem ser usados como ferramentas didáticas, a Robot ARena possibilita que jogos com objetivos educacionais sejam desenvolvidos, fazendo uso das inovações trazidas pela AR. Não somente jogos educacionais podem fazer uso da Robot ARena, mas também outros “*serious games*”, que podem ser aplicados, por exemplo, em publicidade, em pesquisas e como ferramenta de expressão para artistas.

Um resultado importante obtido através da análise dos protótipos de jogos desenvolvidos é a constatação de que não é a simples implementação de AR em jogos que traz inovação à jogabilidade, mas sim a exploração das novas interfaces que a AR disponibiliza e também a utilização dos melhores aspectos presentes nos jogos tradicionais e eletrônicos.

Finalmente pode-se afirmar que este trabalho representa apenas o início do desenvolvimento da Robot ARena, e que esta fornece uma plataforma flexível para diversas outras pesquisas e experimentos envolvendo interfaces horizontais, realidade aumentada espacial, jogos, robôs etc. Algumas destas pesquisas são discutidas na seção seguinte.

7.1 TRABALHOS FUTUROS

Uma possibilidade trazida pela Robot ARena, e pelo jogo FootBot Arena, é o desenvolvimento de aplicações com usuários remotos, conectados em rede, em uma partida, onde cada usuário controla seu próprio robô real, que interage com uma representação virtual do robô adversário. Esta representação virtual pode ser um modelo 3D, mas também pode ser um vídeo-avatar do robô remoto.

Outros dispositivos de entrada podem ser desenvolvidos para a Robot ARena, como, por exemplo um sistema de multi-toques, muito comum em interfaces horizontais, e que envolve pesquisas tanto no reconhecimento dos toques dos usuários, como no tratamento destes toques pela aplicação ou sistema operacional. Na Robot Arena estes sistema de multi-toques pode ser desenvolvido através de reconhecimento de sombras na superfície de projeção ou através da técnica de reflexão interna de luz infravermelha no acrílico, como em (HAN, 2005). Outra forma de entrada que pode ser aplicada é o reconhecimento de gestos dos usuários, como o trabalho desenvolvido no enJine de interação com *desktops* 3D através de gestos (VICENTIM; TOKUNAGA; MUTO, 2007).

Com o objetivo de dar um maior realismo ao ambiente aumentado é possível utilizar projeções estereoscópicas, que aumentam a percepção tridimensional da cena. Porém, este efeito é dependente do ponto de vista do usuário, e para que este possa se movimentar livremente é necessário que seu ponto de vista seja rastreado.

A Robot ARena permite, ainda, o uso das projeções para aplicar texturas diretamente nos objetos reais, aplicando as técnicas de projeções geométricas apresentadas por Bimber e Raskar (2005) e descritas nesta dissertação, alterando, assim, sua aparência visual. Como, por exemplo, mudando a cor do robô para representar o time qual ele pertence, ou adicionando efeitos especiais, como o robô

piscando em vermelho, avisando que está prestes a explodir. Outra possibilidade é a iluminação direcionada dos objetos reais, permitindo que o ambiente seja escurecido e os objetos reais fiquem tão realçados quanto os objetos virtuais.

Por fim, um importante trabalho futuro é o desenvolvimento de aplicações e jogos com objetivos educacionais e a análise destes jogos verificando sua efetividade como recurso educacional e também da Robot ARena como ferramenta para o desenvolvimento destes aplicativos. Acredita-se que a Robot ARena possa ser uma infra-estrutura eficiente nesta área, visto a vocação educacional dos elementos que a compõem, como o enJine e o Lego, que possuem resultados comprovados nessa área.

REFERÊNCIAS

AOKI, T.; ICHIKAWA, H.; ASANO, K.; MITAKE, H.; IIO, Y.; AYUKAWA, R.; KURIYAMA, T.; KAWA, T.; MATSUMURA, I.; MATSUSHITA, T.; TOYAMA, T.; HASEGAWA, S.; SATO, M. Kobito - Virtual Brownie: Virtual creatures interact with real objects and real people. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'2005 Emerging Technologies), 2005, Los Angeles. **Anais...** New York: ACM Press, 2005.

ARTOOLKIT Homepage. Disponível em: <<http://www.hitl.washington.edu/artoolkit/>>. Acesso em: 24 abr. 2007.

AZUMA, R. T. A Survey of Augmented Reality. **Presence: Teleoperators and Virtual Environments**, v.6, 4, p. 355-385, ago. 1997.

AZUMA, R. T.; BAILLOT, Y.; BEHRINGER, R.; FEINER, S. K.; JULIER, S.; MACINTYRE, B. Recent Advances in Augmented Reality. **IEEE Computer Graphics and Applications**, v.21, 6, p. 34-47, nov./dez. 2001.

BAUER, M.; BRUEGGE, B.; KLINKER, G.; MACWILLIAMS, A.; REICHER, T.; RIB, S.; SANDOR, C.; WAGNER, M. Design of a Component-Based Augmented Reality Framework. In: Proceedings of the IEEE and ACM International Symposium on Augmented Reality (ISAR'2001), 2001, New York. **Anais...** Washington: IEEE Computer Society, 2001, p. 45-54.

BERNARDES JR., J.L. **Desenvolvimento de um ambiente para visualização tridimensional da dinâmica de risers**. 2004. 203 p. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2004.

BERNARDES JR., J.L. **Questões Importantes** [mensagem pessoal]. Mensagem recebida por <calife@gmail.com> em 24 nov. 2006.

BERNARDES JR., J.L.; DIAS, J.S.; TORI, R. Exploring Mixed Reality User Interfaces for Electronic Games. In: Anais do IV Workshop Brasileiro de Jogos e Entretenimento Digital (WJOGOS'2005), 4., 2005, São Paulo. **Anais...** Porto Alegre: SBC, 2005, p. 353-358.

BERNARDES JR, J.L.; NAKAMURA, R.; TORI, R. Using a Didactic Game Engine to Teach Computer Science. In: Digital Proceedings of the VI Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES'07 Tutorials), 6., 2007, São Leopoldo. **Anais...** Porto Alegre: SBC, 2007. Disponível em: <<http://www.inf.unisinos.br/~sbgames/anais/tutorials/index.html>>. Acesso em: 31 dez. 2007.

BIANCHINI, R.C. **Uma Arquitetura BDI para Comportamentos Interativos de Agentes em Jogos Computacionais**. 2005. 236 p. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2005.

BIANCHINI, R.C.; BERNARDES JR., J.L.; CUZZIOL, M.; JACOBBER, E.; NAKAMURA, R.; TORI, R. Jogos Eletrônicos e Realidade Virtual. In: TORI, R.; KIRNER, C.; SISCOOTTO, R. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Porto Alegre: SBC, 2006, 2, p. 199-219.

BILLINGHURST, M.; GRASSET, R.; LOOSER, J. Designing Augmented Reality Interfaces. **ACM SIGGRAPH Computer Graphics**, v.39, 1, p. 17-22, fev. 2005.

BIMBER, O.; RASKAR, R. **Spatial Augmented Reality: Merging Real and Virtual Worlds**. Wellesley: A K Peters, 2005. 369 p.

BIMBER, O.; RASKAR, R. **Modern Approaches to Augmented Reality**. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'2006 Courses), 2006, Boston. **Anais...** New York: ACM Press, 2006.

BLOW, J. Game Development: Harder Than You Think. **ACM Queue**, v.1, 10, p. 28-27, fev. 2004.

BLUECOVE HomePage. Disponível em: <<http://bluecove.sourceforge.net/>>. Acesso em: 31 dez. 2007.

BRAVE, S.; ISHII, H.; DAHLEY, A. Tangible interfaces for remote collaboration and communication. In: Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work, 1998, Seattle. **Anais...** New York: ACM Press, 1998, p. 169-178.

BRYSON, S.; ZELTZER, D.; BOLAS, M.T.; CHAPELLE, B.; BENNETT, D. The Future of Virtual Reality: Head Mounted Displays versus Spatially Immersive Displays. In: Proceedings of the 23rd Annual Conference on Computer Graphics and

Interactive Techniques (SIGGRAPH'96), 23., 1996, New Orleans. **Anais...** New York: ACM Press, 1996, p. 485-486.

BURROUGHES, J.H.; BRADLEY, D.C.; BROWN, A.R.; MARKS, R.N.; MACKAY, K.; FRIEND, R.H.; BURNS, P.L.; HOLMES, A.B. Light Emitting Diodes Based on Conjugated Polymers. **NPG Nature**, 347, p. 539-541, out. 1990.

BUSHNELL, N. Relationships Between Fun and Computer Business. **Communications of the ACM**, v.39, 8, p. 31-37, ago. 1996.

CAILLOIS, R. **Les Jeux et les Hommes**. Paris: Gallimard, 1958, 306p.

CALIFE, D.; BERNARDES JR, J.L.; TORI, R. Robot ARena: an Augmented Reality Platform for Game Development. In: Proceedings of the VI Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES'2007), 6., 2007, São Leopoldo. **Anais...** Porto Alegre: SBC, 2007, p. 77-86.

CALIFE, D.; TOMOYOSE, A.; SPINOLA, D.; BERNARDES JR., J.L.; TORI, R. Robot ARena: Infraestrutura for Applications Involving Spatial Augmented Reality and Robots. In: Proceedings of the IX Symposium on Virtual and Augmented Reality (SVR'2007), 9., 2007, Petrópolis. **Anais...** Porto Alegre: SBC, 2007.

CALIFE, D.; TOMOYOSE, A.; SPINOLA, D.; TORI, R. Controle e Rastreamento de um Robô Real para um Ambiente de Realidade Misturada. In: Anais do II Workshop de Aplicações de Realidade Virtual (WARV'2006), 2., 2006, Recife. **Anais...** Porto Alegre: SBC, 2006.

CHEOK, A.D.; GOH, K.H.; LIU, W.; FARBIZ, F.; FONG, S.W.; TEO, S.L.; LI, Y.; YANG, X. Human Pacman: a Mobile, Wide-Area Entertainment System Based on Physical, Social, and Ubiquitous Computing. **Personal and Ubiquitous Computing**, v.8, 2, p. 71-81, mai. 2004.

CRAWFORD, C. **Chris Crawford on Game Design**. Berkeley: New Riders Publishing, 2003, 496p.

DOWNING, A.; HESSELINK, L.; RALSTON, J.; MACFARLANE, R. A Three-Color, Solid-State, Three-Dimensional Display. **AAAS Science**, v. 273, 5279, p. 1185-1189, ago. 1996.

ENJINE: Engine for Games in Java Homepage. Disponível em: <<http://www.enjine.org/>>. Acesso em: 20 nov. 2007.

EYETOY Homepage. Disponível em: <<http://www.eyetoy.com/index.asp>>. Acesso em: 25 abr. 2007.

FEINER, S.; MACINTYRE, B.; HAUPT, M.; SOLOMON, E. Windows on the World: 2D Windows for 3D Augmented Reality. In: Proceedings of ACM Symposium on User Interface Software and Technology, 1993, Atlanta. **Anais...** New York: ACM Press, 1993, p. 145-155.

GEIGER, C.; REIMANN, C.; STICKLEIN, J.; PAELKE, V. JARTOOLKIT – A Java binding for ARToolKit. In: Proceedings of the First IEEE International Workshop on Augmented Reality Toolkit, 1., 2002. **Anais...** Washington: IEEE Computer Society, 2002, 5pp.

HAN, J.Y. Low-Cost Multi-Touch Sensing Through Frustrated Total Internal Reflection. In: Proceedings of the 18th annual ACM Symposium on User interface Software and Technology (UIST'05), 18., 2005, Seattle. **Anais...** New York: ACM Press, 2005, p. 115-118.

HOWARD, W.E.; PRACHE, O.F. Microdisplays Based Upon Organic Light Emitting Diodes. **IBM Journal of Research and Development**, v.45, 1, 2001. Disponível em: <<http://www.research.ibm.com/journal/rd/451/howard.html>>. Acesso em: 25 abr. 2007.

HUIZINGA, J. **Homo Ludens: o Jogo como Elemento da Cultura**, 5ª edição. São Paulo: Perspectiva, 2004. 243p.

ISHII, H.; ULLMER, B. Tangible Bits: towards seamless interfaces between people, bits and atoms. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1997, Atlanta. **Anais...** New York: ACM Press, 1997, p. 234-241.

JACOBBER, E.C. **Proposta e Implementação de uma Interface para Motores de Jogos Interativa e Centrada no Usuário**. 2007. 170 p. Dissertação (Mestrado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

JACOBSON, I; BOOCH, G.; RUMBAUGH, J. **The Unified Software Development Process**. Boston: Addison-Wesley, 1999. 463 p.

JAVA Communications API Homepage. Disponível em: <<http://java.sun.com/products/javacomm/>>. Acesso em: 20 abr. 2007.

KIRNER, C.; TORI, R. Fundamentos de Realidade Aumentada. In: TORI, R.; KIRNER, C.; SISCOOTTO, R. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Porto Alegre: SBC, 2006, 2, p. 22-38.

LASER Magic Productions Homepage. Disponível em: <<http://www.laser-magic.com/>>. Acesso em: 25 abr. 2007.

LEGO Mindstorms Homepage. Disponível em: <<http://mindstorms.lego.com/>>. Acesso em: 20 abr. 2007.

LEJOS Java for Lego Mindstorms HomePage. Disponível em: <<http://lejos.sourceforge.net/>>. Acesso em: 31 dez. 2007.

LEWIS, M.; JACOBSON, F. **Game Engines in Scientific Research**. Communications of the ACM, v.45, 1, p. 27-31, jan. 2002.

LOW, K.; WELCH, G.; LASTRA, A.; FUCHS, H. Life-Sized Projector-Based Dioramas. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 2001, Baniff. **Anais...** New York: ACM Press, 2001, p. 93-101.

MACWILLIAMS, A.; SANDOR, C.; WAGNER, M.; BAUER, M.; KLINKER, G.; BRUEGGE, B. Herding Sheep: Live System Development for Distributed Augmented Reality. In: Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'2003), 2., 2003, Tokyo. **Anais...** Washington: IEEE Computer Society, 2003, p. 123-132.

MAGERKURTH, C.; CHEOK, A.D.; MANDRYK, R.L.; NILSEN, T. Pervasive Games: Bringing Computer Entertainment Back to the Real World. **ACM Computer in Entertainment**, v.3, 3, p. 1-19, jul. 2005.

MAGERKURTH, C.; ENGELKE, T.; MEMISOGLU, M. Augmenting the Virtual Domain with Physical and Social Elements: Towards a Paradigm Shift in Computer Entertainment Technology. In: Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, 2004, Singapura. **Anais...** New York: ACM Press, 2004, p. 163-172.

MILGRAM, P.; TAKEMURA, H.; UTSUMI, A.; KISHINO, F. Augmented Reality: a class of displays on the reality-virtuality continuum. **SPIE Telemanipulator and Telepresence Technologies**, v.2351, p. 282-292, 1994.

NILSEN, T.; LINTON, S.; LOOSER, J. Motivations for Augmented Reality Gaming. In: Proceedings of the New Zealand Game Developers Conference (FUSE'04), 2004, Dunedin. **Anais...** [S.l.:s.n.] 2004, p. 86-93.

NINTENDO Wii Homepage. Disponível em: <<http://wii.nintendo.com/>>. Acesso em: 25 abr. 2007.

OPENCV Open Source Computer Vision Library Homepage. Disponível em: <<http://www.intel.com/technology/computing/opencv/>>. Acesso em: 20 abr. 2007.

PIEKARSKY, W.; THOMAS, B. ARQuake: the Outdoor Augmented Reality Gaming System. **Communications of the ACM**, v.45, 1, p. 36-38, jan. 2002.

PURDY, J.A. Serious Games: Getting Serious About Digital Games in Learning. **Corporate University Journal**. Disponível em: <<http://www.corpu.com/newsletter%5Fwi07/sect2.asp>>. Acesso em: 23 abr. 2007.

RASKAR, R.; BEARDSLEY, P.; BAAR, J.; WANG, Y.; DIETZ, P.; LEE, J.; LEIGH, D.; WILLWACHER, T. RFIG Lamps: Interacting with a Self-Describing World via Photosensing Wireless Tags and Projectors. In: Proceedings International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'2004), 2004, Los Angeles. **Anais...** New York: ACM Press, 2004, p. 406-415.

RASKAR, R.; LOW, K. Interacting with Spatially Augmented Reality. In: Proceedings of the 1st International Conference on Computer Graphics, Virtual Reality and Visualization, 1., 2001, Camps Bay. **Anais...** New York: ACM Press, 2001, p. 101-108.

RASKAR, R.; WELCH, G.; FUCHS, H. Spatially Augmented Reality. In: First IEEE International Workshop on Augmented Reality (IWAR'98), 1., 1998, San Francisco. **Anais...** San Francisco: A.K. Peters, 1998, p. 1-7.

RHYNE, T.M. Computer Games and Scientific Visualization. **Communications of the ACM**, v.45, 7, p. 40-44, jul. 2002.

ROBOCUP HomePage. Disponível em: <<http://www.robocup.org/>>. Acesso em: 31 dez. 2007.

ROLLINGS, A.; ADAMS, E. **Andrew Rollings and Ernest Adams on Game Design**. Berkeley: New Riders, 2003. 648 p.

ROSENBLOOM, A. A Game Experience in Every Application. **Communications of the ACM**, v.46, 7, p. 28-31, jul. 2003.

RXTX Serial and Parallel I/O Libraries Supporting Sun's CommApi HomePage. Disponível em: <<http://www.rtx.org/>>. Acesso em: 31 dez. 2007.

SALEN, K.; ZIMMERMAN, E. **Rules of Play: Game Design Fundamentals**. Cambridge: The MIT Press, 2004. 688 p.

SHEN, C.; VERNIER, F.D.; FORLINES, C.; RINGEL, M. DiamondSpin: an extensible toolkit for around the table interaction. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'2004), 2004, Vienna. **Anais...** New York: ACM Press, 2004, p. 167-174.

SMITH, R. **Game Impact Theory: The Five Forces that are Driving the Adoption of Game Technologies within Multiple Established Industries**. Disponível em: <<http://www.ctonet.org/journals.htm>>. Acesso em: 18 fev. 2007.

STARNER, T.; LEIBE, B.; SINGLETARY, B.; PAIR, J. MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Game. In: Proceedings of the 5th International Conference on Intelligent User Interfaces, 2000, New Orleans. **Anais...** New York: ACM Press, 2000, p. 256-259.

STUDIERSTUBE Augmented Reality Project Homepage. Disponível em: <<http://studierstube.icg.tu-graz.ac.at/>>. Acesso em: 25 abr. 2007.

SUGIMOTO, M.; KOJIMA, M.; NAKAMURA, A.; KAGOTANI, G.; NII, H.; INAMI, M. Augmented Coliseum: Display-Based Computing for Augmented Reality Inspiration Computing Robot. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'2005 Emerging Technologies), 2005, Los Angeles. **Anais...** New York: Acm Press, 2005.

SWARTOUT, W.; VAN LENT, M. Making a Game of System Design. **Communications of the ACM**, v.46, 7, p. 32-39, jul. 2003.

TORI, R.; BERNARDES JR, J. L.; NAKAMURA, R. Teaching Introductory Computer Graphics Using Java 3D, Games and Customized Software: a Brazilian Experience. In: Proceedings of the International Conference on Computer Graphics and Interactive Techniques (ACM SIGGRAPH'2006 Educators Program), 2006, Boston. **Anais...** New York: ACM Press, 2006.

TORI, R.; NAKAMURA, R.; BERNARDES JR. J.L.; BIANCHINI, R.C.; JACOBER, E.C.; CALIFE, D.; TOMOYOSE, A.N. Jogos e Entretenimento com Realidade Virtual e Aumentada. In: TORI, R.; KIRNER, C. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações**. Porto Alegre: SBC, 2007, 11, p. 192-222.

TSUDA, F.; HOKAMA, P.M.; RODRIGUES, T.M.; BERNARDES JR., J.L. Integration of jARToolKit and enJine: extending with AR the potential use of a didactic game engine. In: Proceedings of the IX Symposium on Virtual and Augmented Reality (SVR'2007), 9., 2007, Petrópolis. **Anais...** Porto Alegre: SBC, 2007.

ULBRICHT, C.; SCHMALSTIEG, D. Tangible Augmented Reality for Computer Games. In: Proceedings of the Visualization, Imaging and Image Processing (VIIP'2003), 2003, Benalmádena. **Anais...** Calgary: ACTA Press, 2003, p. 950-954.

UNDERKOFFLER, J.; ISHII, H. Illuminating Light: an optical design tool with a luminous tangible interface. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 1998, Los Angeles. **Anais...** New York: ACM Press, 1998, p. 542-549.

VAN DEN BERGH, F.; LALIOTI, V. Software Chroma Keying in a Immersive Virtual Environment. **South African Computer Journal**, 24, p. 155-162, 1999.

VICENTIM, A.P.; TOKUNAGA, D.M.; MUTO, M.M. **Desenvolvimento e Estudo de um Desktop3D com o Uso de Gestos**. 2007. 100 p. Monografia (Trabalho de Formatura) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

VIEIRA, B.N.S.; THEODORO, C.; TRIAS, L.P.; MIRANDA, F.R.; TORI, R. ARHockey: Um Jogo em Realidade Aumentada Baseada em Projetores. In: Digital Proceedings of the V Brazilian Symposium on Computer Games and Digital Entertainment (SBGAMES'2006), 5., 2006, Recife. **Anais...** Porto Alegre: SBC, 2006.

WALTHER, B.K. Playing and Gaming: Reflections and clarifications. **Game Studies: the international journal of computer game research**, v.3, 1, mai. 2003. Disponível em: <<http://www.gamestudies.org>>. Acesso em: 25 abr. 2007.

WILSON, A.D. PlayAnywhere: a compact interactive tabletop projection-vision system. In: Proceedings of the 18th annual ACM Symposium on User interface Software and Technology (UIST'05), 18., 2005, Seattle. **Anais...** New York: ACM Press, 2005, p. 83-92.

WITTGENSTEIN, L. **Philosophical Investigations**, 3ª edição. New Jersey: Prentice Hall, 1999, 250p.

WU, M.; BALAKRISHNAN, R. Multi-Finger and Whole Hand Gestural Interaction Techniques for Multi-User Tabletop Displays. In: Proceedings of the 16th annual ACM Symposium on User Interface Software and Technology (UIST'03), 16., 2003, Vancouver. **Anais...** New York: ACM Press, 2003, p. 193-202.

XEROX Parc Homepage. Electronic Reusable Paper. Disponível em: <<http://www2.parc.com/hsl/projects/gyricon/>>. Acesso em: 25 abr. 2007.

ZYDA, M.; HILES, J.; MAYBERRY, A.; WARDYNSKI, C.; CAPPS, M.; OSBORN, B.; SHILLING, R.; ROBASZEWSKI, M.; DAVIS, M. Entertainment R&D for Defense. **IEEE Computer Graphics and Applications**, v.23, 1, p. 28-36, jan./fev. 2003.

GLOSSÁRIO

Advergames: São jogos que têm como objetivo passar uma mensagem publicitária, promover uma marca, produto ou associar uma experiência a estes produtos e marcas.

Avatar: Personagem ou objeto virtual que representa um jogador, ou um agente em um ambiente virtual.

Câmera de Orifício: Uma câmera com um orifício extremamente pequeno que pode focar a luz dentro de sua caixa escura, captando os raios de luz através de seu orifício.

Código-Gray: Sistema numérico binário, onde dois valores consecutivos diferem em apenas um bit.

Design de Jogos: Processo de projetar as regras, experiências e conteúdo de um jogo.

Jogo de Estratégia em Tempo Real: Jogo de estratégia onde as ações dos jogadores não são organizadas por turnos, como em jogos de estratégia tradicionais. Cada jogador pode tomar suas decisões a qualquer momento no jogo.

HMD: Dispositivo de exibição acoplado à cabeça de usuário.

Jittering: Variação indesejada de alguma característica de um sinal, normalmente aplicada em eletrônica e tele-comunicações.

Jogabilidade: São as experiências obtidas por um jogador ao interagir com um jogo, normalmente não levando em conta especificidades da parte gráfica, som e a história. Em alguns casos pode ser definido como o que o jogador faz no jogo.

Jogos de Aventura: Jogos tradicionais em computadores que também podem ser considerados como uma história interativa. O objetivo do jogo é se aventurar e interagir com uma história em um ambiente virtual, envolvendo exploração, investigação e resolução de quebra-cabeças.

Marcador Fiducial: objeto com propriedades e características conhecidas, normalmente utilizado em visão computacional.

Mecânicas de Jogo: Conjunto de regras com o objetivo de montar um jogo ou jogabilidade.

Optical Flow: Conceito que aproxima movimentos de objetos em representações visuais.

Role-Playing Game: Jogo de interpretação de personagens, onde um jogador pode interpretar personagens que interagem e evoluem em seus mundos de acordo com as regras do jogo.

Sistema Multi-Toques: Sistema de interface humano-computador que permite a interação do usuário através de múltiplos toques simultâneos diretamente no dispositivo de exibição.

Tiro em Primeira Pessoa: Tipo de jogo de ação onde o jogador tem a visão do jogo através de uma perspectiva em primeira pessoa, cujo o objetivo é matar os inimigos com diferentes tipos de armas.

Transponder: Um transmissor/receptor que envia um sinal de resposta à um sinal específico recebido.

Vídeo-Avatar: Avatar representado através de um vídeo em tempo real, como por exemplo, um vídeo de um jogador como seu próprio avatar em um jogo.

Voxel: Elemento de volume, representa um valor em um espaço tridimensional.

Widgets: Elementos de uma interface gráfica de usuário.

APÊNDICE A – DOCUMENTAÇÃO DO PROJETO

Neste apêndice é apresentada a documentação gerada durante o projeto da Robot ARena.

Casos de Uso

Aqui são apresentados os casos de uso analisados durante o projeto, estes casos de uso foram construídos sob o ponto de vista de desenvolvimento de um jogo na Robot ARena e de sua utilização.

Atores

Desenvolvedor do Jogo: Este ator representa uma pessoa que está utilizando a infra-estrutura para desenvolver um jogo específico. Assume-se que o desenvolvedor do Jogo tenha conhecimentos de programação e da linguagem Java.

Usuário do Jogo: Este ator representa um usuário do jogo, ou seja, um jogador.

Casos de Uso

Criação de um Robô Real Rastreado

Atores: Desenvolvedor do jogo.

Descrição: Este caso de uso se inicia quando o desenvolvedor decide criar um robô real rastreado em seu jogo. Dentro do jogo um robô real atua como um robô virtual, apenas não possui uma representação visual, portanto o desenvolvedor deve definir a estrutura lógica do robô real. Outra diferença de um objeto virtual é que a estrutura lógica do robô real não é atualizada diretamente pelo usuário, o sistema de rastreamento deve atualizar as propriedades do robô real. O próximo passo do

desenvolvedor é atribuir um dispositivo de rastreamento à estrutura lógica do robô real.

Definição da Comunicação e Controle do Robô Real

Atores: Desenvolvedor do jogo.

Descrição: Este caso de uso de inicia quando desenvolvedor decide criar um canal de comunicação entre o software e o robô, permitindo seu controle ao usuário ou a outros elementos do jogo. De acordo com a tecnologia utilizada para montar o robô real o sistema de comunicação deve permitir o controle de movimentos básicos do robô como: mover para frente, mover para trás, virar a direita, virar a esquerda e parar.

Criação de uma Interface Tangível

Atores: Desenvolvedor do jogo.

Descrição: Este caso de uso se inicia quando o desenvolvedor decide criar um elemento de entrada do usuário tangível. Dentro do jogo, este elemento pode possuir uma representação visual, portanto o primeiro passo do desenvolvedor é criar a estrutura lógica deste elemento e sua representação visual, se houver, que nada mais é do que um modelo geométrico de um sólido, criado através de ferramentas adequadas.

No segundo passo, o desenvolvedor deve definir um dispositivo de rastreamento ao elemento tangível, para que as entradas do usuário, que manipula a representação real deste elemento, possam ser transmitidas ao jogo.

Movimentação do Robô Real

Atores: Usuário do jogo.

Descrição: Este caso de uso de se inicia quando o usuário envia um comando de movimento ao robô real. O sistema deve verificar se a representação lógica do robô não está colidindo com algum elemento virtual do jogo, caso não esteja colidindo o sistema envia, através do sistema de comunicação com robô, o comando do usuário.

Após a movimentação a posição lógica do robô deve ser atualizada pelo sistema de rastreamento, e caso haja alguma colisão com um elemento virtual a resolução desta colisão deve ser efetuada.

Controle da Interface Tangível

Atores: Usuário do jogo.

Descrição: Este caso de uso se inicia quando o usuário manipula o elemento de interação da interface tangível, com o objetivo de enviar um comando de entrada ao jogo. O sistema de rastreamento deve atualizar a posição lógica do elemento de interação, assim as entradas do usuário podem ser reconhecidas.

Efetua Calibragem da Projeção

Atores: Usuário do jogo.

Descrição: Este caso de uso se inicia quando o usuário decide iniciar a projeção de um jogo e deve calibrar esta projeção adequadamente. Na superfície de projeção é apresentado um padrão visual, que pode ser utilizado pelo usuário, calibragem manual, ou por um sistema de visão computacional, calibragem automática. O usuário pode decidir por ambas abordagens, inclusive em conjunto. Na calibragem automática o usuário envia um comando que inicia o sistema de visão computacional, que reconhece o padrão e modifica os parâmetros da câmera virtual do jogo. Na calibragem manual o usuário envia comandos de entrada que modificam diretamente os parâmetros da câmera virtual. Logo após os ajustes o usuário envia um comando para que o sistema grave os parâmetros de calibragem em um arquivo.

Diagramas de Casos de Uso

Diagrama de Componentes

O diagrama abaixo mostra os subsistemas planejados para a Robot ARena e suas relações.

Diagrama de Classes

Abaixo estão os diagramas de classes, que embora já apresentados, aqui são reproduzidos em tamanho ampliado.

Subsistema de Visualização

Subsistema de Rastreamento e Registro

Subsistema de Elementos Reais

Subsistema de Comunicação e Controle do Robô