

FICHA CATALOGRÁFICA

Dados Internacionais de Catalogação na Publicação (CIP)

G633i Gomes, Wnêiton Luiz, 1980 -
 Uma interface gráfica com o usuário de apoio ao software ARToolKit /
 Wnêiton Luiz Gomes. - 2007.
 68 f. : il.
 Orientador: Alexandre Cardoso.
 Co-orientador: Edgard Lamounier Jr.
 Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
 Inclui bibliografia.
 1. Realidade virtual - Teses. I. Cardoso, Alexandre. II. Lamounier Junior, Edgar. III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 681.3:007.52

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

WNÊITON LUIZ GOMES



Uma Interface Gráfica com o Usuário de Apoio ao Software ARToolKit

Orientador:

Prof^o. Alexandre Cardoso, Dr

Co-orientador:

Prof^o. Edgard Lamounier Jr, PhD

Uberlândia, MG, julho de 2007.

WNÊITON LUIZ GOMES

Uma Interface Gráfica com o Usuário de Apoio ao Software ARToolKit

Dissertação apresentada por Wnêiton Luiz Gomes, à
Universidade Federal de Uberlândia (UFU),
Faculdade de Engenharia Elétrica, para obtenção do
título de Mestre em Ciências, aprovada em
27/07/2007 pela banca examinadora.

Área de Concentração

Processamento da Informação

Banca Examinadora:

Profº. Alexandre Cardoso, Dr (UFU)

Profº. Adriano Alves Pereira, Dr (UFU)

Profº. José Remo Ferreira Brega, Dr (UNESP)

Uberlândia, MG, Julho de 2007.

Uma Interface Gráfica com o Usuário de Apoio ao Software ARToolKit

Dissertação apresentada por Wnêiton Luiz Gomes, à
Universidade Federal de Uberlândia (UFU),
Faculdade de Engenharia Elétrica, para obtenção do
título de Mestre em Ciências.

Alexandre Cardoso, Dr
Orientador

Darizon A. Andrade, PhD
Coordenador

Uberlândia, MG, Julho de 2007.

“Bem aventurado o homem que acha a
sabedoria e o homem que adquire conhecimento”

(Prov. 3:13)

AGRADECIMENTOS

Aos meus orientadores, Professores Alexandre Cardoso e Edgard Lamounier, pela orientação prestada nos momentos da realização deste trabalho.

Aos meus amigos do laboratório de Computação Gráfica, Marlene Marques, Arquimedes, Maria Emília, Marlene Roque, Naira, Mônica Rocha, Afonso, Gleison, Leonardo, Luiz Fernando, pela troca de conhecimento, experiências e pela amizade.

Uma dedicação exclusiva para o Luciano Ferreira Silva e ao Ezequiel Zorzal pela excelente ajuda na estrutura da dissertação e pela amizade.

Ao meu amigo e colega Kenedy Lopes, parceiro de trabalho me ajudou muito na realização deste trabalho.

Ao Elvio do laboratório Qualidade de Energia pela ajuda prestada no desenvolvimento do software.

Aos meus colegas de república, Ezequiel Zorzal e Arthur Bastos, o Ezequiel pelas panelas lavadas e ao Arthur por me ajudar no preparo das refeições.

Aos meus primos, Derly e Rosália por ter me recebido e me acolhido numa cidade onde não conhecia ninguém.

Aos meus pais José Luiz Gomes Filho e Elena Miguel do Carmo Gomes, além de meus irmãos, Carlos, Renato, Eliane e José Luiz, por acreditarem na realização deste trabalho.

Agradeço também à UFU e a Capes por ajudar a viabilizar meus estudos através da concessão de uma bolsa de estudos.

Por último e não menos importante, agradeço a Deus por tudo que ele me concedeu para a realização desta obra.

RESUMO

Gomes, L. W. Uma Interface Gráfica com o Usuário de Apoio ao Software ARToolKit. Uberlândia, 2007. 62p. Dissertação (Mestrado em Ciências) Programa de Pós-Graduação em Engenharia Elétrica, UFU, 2007.

Palavras-Chave: Interfaces, Realidade Aumentada, ARToolKit.

Atualmente, a Realidade Virtual e a Realidade Aumentada tem se destacado devido a suas inúmeras aplicações em variados campos. Áreas como educação, indústria, arquitetura, engenharias e entretenimento estão entre as mais beneficiadas.

Em particular, a área de Realidade Aumentada (RA) permite a visualização de informações de forma muito natural, devido sua característica de misturar o virtual com o real. Para a mistura de objetos virtuais com o mundo real, uma das formas é o uso da biblioteca ARToolKit, baseada em técnicas de visão computacional. Contudo, para a utilização do mesmo, qualquer usuário necessita realizar uma seqüência de passos não triviais, envolvendo a abertura de arquivos, sua associação com outros arquivos e a correta colocação dos mesmos em diretórios específicos e conseqüentemente dificultando a utilização da ferramenta pelos usuários.

Visando suprir tais dificuldades, este trabalho apresenta o desenvolvimento de um sistema de janelas para o software ARToolKit, que permite ao usuário a concepção, inserção e associação de marcadores a objetos virtuais de uma maneira mais interativa. Esta interface pode ser utilizada, sem dificuldades por usuários comuns no desenvolvimento, de diversas aplicações em RA.

Além disso, foi feito um estudo, onde se mostram princípios de interação humano-computador, interfaces e usabilidade. Também são apresentados alguns trabalhos no meio científico, principalmente aqueles que se relacionam com o uso de interfaces para a RA.

Em relação aos aspectos tecnológicos, explorou-se a biblioteca ARToolKit, a fim de se saber qual a melhor ferramenta a ser utilizada para a construção de uma interface interativa que possa integrar com a biblioteca.

De acordo com as avaliações dos usuários, o sistema proposto atinge seus objetivos, por ser um sistema de apoio ao usuário no desenvolvimento de aplicações em RA de uma maneira mais intuitiva.

ABSTRACT

Gomes, L. W. A Graphic Interface with the User of Support to the Software ARToolKit. Uberlândia, 2007. 62p. Dissertation (Master's degree in Sciences) Program of Masters degree in Electric Engineering, UFU, 2007.

Key-words: Interfaces, Augmented Reality, ARToolKit.

In these days, the Virtual and Augmented Reality have outstanding due their countless applications in varied fields. Areas as education, industry, architecture, engineering and entertainment are the most beneficiaries.

In particular, the area of Augmented Reality allows the information visualization in a natural form, due from the characteristic of mixing the virtual with the real. For the mixture of virtual objects with the real world, one in the ways is the use of the ARToolKit library was developed, based on techniques of computational vision. However, for its use, any user needs to accomplish a sequence of non trivial steps, involving the opening of files, association with other files and the correct placement of them in specific directories, and consequently hindering the use of the tool for the users.

Seeking to supply such difficulties, this work presents the development of a system of windows for the ARToolKit software that allows to the user the conception, insert and association of markers to virtual objects in a more interactive way. This interface can be used without difficulties for common users in the development of several applications in Augmented Reality.

Besides, a study was made showing principals of human-computer interaction, interfaces and usability. Some works in the scientific area are also presented, mainly those that link with the use of interfaces for Augmented Reality.

In relation to the technological aspects the ARToolKit library was explored, in order to know which tool is the best to be used for the construction of an interactive interface that can integrate with the library.

In agreement with the users' evaluations, the proposed system reaches its objectives, by a system for user support in the development of Augmented Reality applications in a more intuitive way.

SUMÁRIO

1.	INTRODUÇÃO	9
1.1.	Motivação	9
1.2.	Objetivos.....	11
1.3.	Estrutura da Dissertação.....	11
2.	FUNDAMENTOS TECNOLÓGICOS.....	12
2.1.	Introdução	12
2.2.	Interação humano-computador	12
2.3.	Definição de Interface e Sua Evolução.....	13
2.4.	O Usuário e a Interface.....	14
2.5.	Fundamentos de RV e RA.....	18
2.6.	Etapas de utilização do ARToolKit.....	22
2.7.	Considerações Finais	26
3.	TRABALHOS RELACIONADOS	27
3.1.	Introdução	27
3.2.	AR System	27
3.3.	AR Pad.....	29
3.4.	Tiles System.....	30
3.5.	Considerações Finais	31
4.	ESPECIFICAÇÃO DO SISTEMA PROPOSTO	33
4.1.	Introdução	33
4.2.	Tecnologias de Apoio	33

4.3.	GLUT e GLUI	33
4.4.	Arquitetura de Módulos do Sistema.....	34
4.5.	GUI – Interface Gráfica	35
4.6.	Módulo Create Marker	35
4.7.	Módulo Register Marker.....	36
4.8.	Módulo Associate Marker	37
4.9.	Especificação do Sistema	38
4.10.	Diagrama de Casos de Usos	39
4.11.	Diagrama de Classes	40
4.12.	Considerações Finais	42
5.	DETALHES DE IMPLEMENTAÇÃO.....	43
5.1.	Introdução	43
5.2.	Implementação do módulo Criador de Marcador	43
5.3.	Implementação do módulo Cadastrador de Marcador	44
5.4.	Implementação do módulo Associador de Marcador	46
5.5.	Considerações Finais	50
6.	RESULTADOS E LIMITAÇÕES	51
6.1.	Introdução	51
6.2.	Utilização do Sistema	51
6.3.	Avaliação do Sistema	51
6.4.	Satisfação quanto ao uso do sistema.....	52
6.5.	Quanto à manipulação de marcadores e objetos virtuais.....	54
6.6.	Limitações do Sistema	56

6.7.	Considerações Finais	56
7.	CONCLUSÕES	57
7.1.	Trabalhos Futuros	58
	REFERÊNCIAS BIBLIOGRÁFICAS	59
	ANEXO	61

LISTA DE FIGURAS

Figura 1. Processo de interação humano-computador.....	12
Figura 2. Diagrama relacionado com o projeto de interface de usuário.....	15
Figura 3. Exemplo de usabilidade na WWW	17
Figura 4. Continuum de Milgram.....	19
Figura 5. Análise da imagem pelo ARToolKit.....	21
Figura 6. Processamento de imagens do ARToolKit.	21
Figura 7. Estrutura ARToolKit.....	22
Figura 8. Conteúdo do arquivo .dat.....	23
Figura 9. Exemplo de marcador.	23
Figura 10. Cadastramento de marcador.....	24
Figura 11. Conteúdo do arquivo vrml_data.....	24
Figura 12. Arquivo vrml_data alterado	25
Figura 13. Diagrama de associações.....	25
Figura 14. Sistema de RA.....	27
Figura 15. Interface AR System	28
Figura 16. Spaceball e handheld.....	29
Figura 17. Ambiente Colaborativo	29
Figura 18. Usuários manipulando dados em um ambiente colaborativo do Tiles System.....	30
Figura 19. Marcadores de Controle	31
Figura 20. Utilização do marcador delete.....	31
Figura 21. Exemplo de objeto virtual e painel de controle.....	34
Figura 22. Arquitetura do Sistema.....	34
Figura 23. Tela inicial do Sistema	35
Figura 24. Interface Create Marker.	36
Figura 25. Interface Register Marker.	37
Figura 26. Tela responsável pelas associações dos marcadores.....	38
Figura 27. Tela de exibição de aplicações.....	38
Figura 28. Diagrama de casos de usos.....	39
Figura 29. Diagrama de classes e pacotes.	41
Figura 30. Fragmento de código do método imgMouseDown.....	43
Figura 31. Fragmento de código do método imgMouseUp.....	44
Figura 32. Fragmento de código inserido no Mk-Patt.....	45
Figura 33. Fragmento de código inserido no Mk-Patt.....	45

Figura 34. Fragmento de código da função <code>control_cb</code>	46
Figura 35. Imagem do marcador e objeto virtual	47
Figura 36. Trecho de código do método <code>LeArquivo</code>	47
Figura 37. Trecho de código do botão <code>Associate</code>	48
Figura 38. Parte da tela da interface que associa marcadores.	49
Figura 39. Facilidade de uso do sistema.....	52
Figura 40. Controle sobre a atividade.....	53
Figura 41. Possibilidade de visibilização clara do que está sendo executado	53
Figura 42. Redução de trabalhos repetitivos	54
Figura 43. Facilidade de Concepção de um novo marcador.....	54
Figura 44. Facilidade de inserção de marcador	55
Figura 45. Facilidade para associar um objeto virtual em um marcador.....	55
Figura 46. Facilidade para remover um objeto de um marcador.....	56

LISTA DE ABREVIATURAS

API - Application Programming Interface

ARToolKit - Augmented Reality Tool Kit

HMD - Head Mounted Display

IHC - Interação Homem-Computador

OpenGL - Open Graphics Library

RA - Realidade Aumentada

RV - Realidade Virtual

VRML - Virtual Reality Modeling Language

WWW - World Wide Web

GLUI - Graphic Library User Interface

GLUT - Graphic Library User Tool Kit

Capítulo 1

1. INTRODUÇÃO

1.1.Motivação

A informática é utilizada em variadas áreas do conhecimento humano. Contudo, usuários ainda encontram dificuldades no uso do computador ou às vezes perdem muito tempo para desenvolver seus trabalhos cotidianos. O papel de uma ferramenta de trabalho – seja ela de software ou não – é tornar possível ou facilitar a realização de uma tarefa. No entanto, softwares mal projetados têm dificultado o uso do computador como ferramenta produtiva.

A utilização do computador como ferramenta de trabalho, na educação, entretenimento e meio de comunicação, entre outros usos, vem exigindo cada vez mais esforços dos projetistas e desenvolvedores de softwares uma preocupação com a qualidade da interface.

Estudos na área de interação humano-computador (IHC), visam à interação entre o usuário e o computador. Esta interação ocorre por meio de interfaces de usuário definidas por software ou hardware.

Uma interface pode ser conceituada como parte do sistema com o qual o usuário realiza contato através do plano físico, perceptivo e cognitivo (Maddix, 1990).

Muitos usuários já se depararam com alguma interface confusa ou até mesmo frustrante. A interface deve ser vista como uma embalagem de um produto (software). Desse modo, ela deve trazer as seguintes características: simplicidade de uso, clareza, facilidade de aprendizagem. Essas características são essenciais para que o usuário não possa ter problemas posteriormente. (Pressman, 2006).

As primeiras interfaces que surgiram foram baseadas em textos e caracteres fazendo uso da metáfora de uma máquina de escrever, essas interfaces são conhecidas como CUI (*Character-based User Interface*). Posteriormente foram criadas as GUI (*Graphic User Interface*), essas interfaces são fundamentadas em gráficos e desenho. Como exemplo, têm-se as interfaces Windows, que faz uso de metáforas de mesas de trabalho, documentos, botões, janelas etc (Valente, 2004).

Apesar de interessante e de ter bom potencial de uso, as interfaces baseadas em janelas ficam restritas à limitação da tela do monitor e ao uso de representações como menus e ícones. Surge então a Realidade Virtual, como uma nova geração de interface, usando

representações tridimensionais mais próximas da realidade do usuário, onde é permitido ao usuário ultrapassar a barreira da tela, além de possibilitar interações mais naturais (Kirner e Siscoutto, 2007). A Realidade Virtual (RV) pode ser definida como uma técnica avançada de interface, onde o usuário pode realizar imersão, navegação e interação em um ambiente sintético tridimensional gerado por computador, utilizando canais multi-sensoriais (Kirner, 1999). O objetivo dessa tecnologia é recriar ao máximo a sensação de realidade para um indivíduo, levando-o a adotar essa interação como uma de suas realidades temporais.

Uma outra tecnologia que vem se destacando como uma nova forma de interface para o usuário é a Realidade Aumentada. Essa tecnologia permite a sobreposição de objetos virtuais sobre ambientes físicos. Com isso, a cena real é enriquecida com objetos virtuais sobrepostos. A RA se popularizou mais nos anos 2000, com o aparecimento, principalmente, do software ARToolKit (*Augmented Reality Tool Kit*) que utiliza técnicas de visão computacional para desenhar os objetos virtuais no mundo real (ARToolKit, 2005).

Considerando o software ARToolKit, o usuário precisa apenas de um dispositivo de rastreamento, que pode ser uma câmera e alguns marcadores onde os objetos serão desenhados.

Apesar de ser uma biblioteca gratuita e apresentar soluções envolvendo equipamentos baratos, para se desenvolver aplicações com o ARToolKit, é necessário que o usuário execute uma sequência de operações não triviais, relacionadas com, cadastrar um novo marcador no sistema, associar objetos virtuais a esses marcadores, executar a aplicação e criar novos marcadores.

As operações citadas acima, atualmente são realizadas manualmente através da edição e cópia de arquivos para diretórios específicos, ou seja, através do uso de interfaces textuais. Isso tudo requer certa destreza por parte do usuário, dificultando assim a utilização do sistema por muitos usuários potenciais.

Todas as operações necessárias para o desenvolvimento de aplicações no ARToolKit, seriam melhor executadas através de uma interface que pudesse facilitar a utilização da ferramenta pelo usuário.

Ao encerrar as anotações sobre a relevância de uma interface, conclui-se que a motivação para esta pesquisa foi o desenvolvimento de uma nova camada (interface) associada ao software ARToolKit, que pudesse facilitar seu uso e consequentemente a disseminação da ferramenta a usuários potenciais.

1.2. Objetivos

Esta pesquisa objetiva prover uma interface gráfica com o usuário de apoio ao software ARToolKit, visando facilitar o processo de criação, cadastramento, associação e manipulação de objetos virtuais que serão utilizados em RA, de maneira rápida e intuitiva.

1.3. Estrutura da Dissertação

Esta dissertação está estruturada da seguinte forma: primeiramente é apresentada a fundamentação teórica, bem como os aspectos tecnológicos que embasará as pesquisas, após isso, fazem-se a descrição dos módulos da interface desenvolvida, as conclusões e as recomendações para futuros trabalhos.

Resumidamente, o texto está estruturado da seguinte forma:

No Capítulo 1 – Introdução, foram anotadas considerações sobre o avanço e a importância da computação na vida das pessoas, mostrando conceitos sobre Realidade Virtual, Realidade Aumentada e alguns conceitos sobre interfaces. Além disso, foram apresentados os objetivos, metas e contribuições da dissertação, bem como a estrutura da dissertação.

No Capítulo 2 – Fundamentos Tecnológicos, são apresentados os conceitos sobre o software ARToolKit em seu estado atual, bem como sua forma de utilização pelos usuários. Além do mais, são abordados conceitos sobre usabilidade, IHC e interface.

No Capítulo 3 – Trabalhos Relacionados: são apresentados o estado da arte, bem como as aplicações dos trabalhos desenvolvidos e o estado em que se encontram as pesquisas na área de Realidade Aumentada.

No Capítulo 4 - Especificação do Sistema Proposto, apresenta-se os diagramas de acordo com a especificação UML, além de detalhar sua arquitetura e as tecnologias de apoio ao seu desenvolvimento.

No Capítulo 5 - Detalhes de Implementação, é efetuada uma descrição sintética da interface desenvolvida, destacando as tecnologias utilizadas e como foi implementada.

No Capítulo 6 – Resultados e Limitações, aqui é visto como é feita a utilização, bem como a avaliação do sistema, além disso, são aplicados testes com usuários sobre questões de usabilidade do sistema desenvolvido, por fim, são apontados resultados e limitações do sistema.

No Capítulo 7 – Nas conclusões, são apresentados as dificuldades encontradas e apontamentos para trabalhos futuros.

Capítulo 2

2. FUNDAMENTOS TECNOLÓGICOS

2.1. Introdução

Este Capítulo apresenta alguns conceitos sobre interação homem-computador, interfaces gráficas e a biblioteca ARToolKit.

2.2. Interação humano-computador

Como mencionado anteriormente, a IHC, é uma área interdisciplinar, relacionada com a Ciência da Computação e com muitos outros campos de estudo e pesquisa. Esta interação ocorre por meio do desenvolvimento de interfaces com usuário definidas por software ou hardware.

No que se refere ao estudo de interação homem-computador, o usuário é tão importante quanto ao desenvolvimento de novas tecnologias. Por isso, alguns pontos fundamentais devem ser considerados durante um processo de desenvolvimento de interface, tais como: o modo com que o usuário interpreta as informações do sistema e que tipo de usuário vai usar esse sistema. Dessa forma, se faz necessário o investimento em técnicas que permitam ao usuário explorar tecnologias por meio de interfaces simples, visto que, o objetivo final é a satisfação do usuário. A Figura 1, ilustra como é o processo de interação homem-computador.

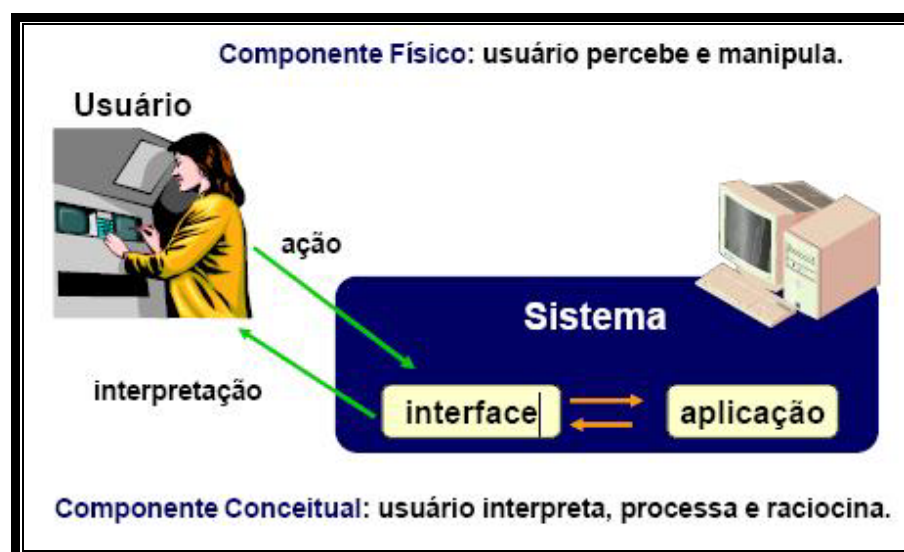


Figura 1. Processo de interação humano-computador

Fonte: (Lima, 2007)

O usuário utiliza a interface como um meio de acesso ao sistema, e este lhe devolve dados ou informações por meio da interface. Com isso, o usuário em seu processo conceitual interpreta e raciocina sobre determinada resposta do sistema.

2.3. Definição de Interface e Sua Evolução

A interface é um mecanismo que faz a junção entre dois modelos: O modelo mental do usuário em relação ao sistema e o modelo de sistema construído pelos engenheiros de software (Norman, 2002). A interface também é definida, como sendo uma fronteira comum entre tecnologias de sistemas de informação, e as maneiras como as informações são extraídas desse sistema. A interface é um conjunto de elementos de hardware e software destinados a possibilitar a interação com o usuário (Ferreira, 1999). Além disso, o termo interface refere-se ao diálogo entre ser humano e o computador.

Segundo definição de (Scapin, 1993), a interface de um software destina-se a interar o usuário com o sistema computacional. Ela é considerada como concernente a todos os aspectos dos sistemas informáticos que influenciam a participação do usuário nas tarefas informatizadas. É bom lembrar, que esta não se refere somente aos aspectos gráficos da camada imediatamente visível pelo usuário na tela do computador, mas também aos objetivos de interação e ao próprio usuário.

Em uma outra definição, a interface é considerada como um dispositivo que serve de limite comum a duas entidades comunicantes, exprimindo-se por uma linguagem específica, sinal elétrico, movimento, língua natural (Valente, 2004).

Ainda em relação a interfaces, se forem simples, direta, agradáveis e divertidas, farão com que o usuário se esqueça da ferramenta e se concentre no trabalho a ser realizado. Contudo, se mal projetadas, darão mais trabalho, comprometendo assim, sua produtividade (Radfahrer apud Fernandes, 2004). Os Modelos de interface centradas no usuário, só começaram a surgir a partir da década de 80, antes, as interfaces homem-máquina eram muito especializadas, voltadas apenas para profissionais da área. Uma das razões que fez do computador uma das mais importantes ferramentas em nosso cotidiano, deve-se às interfaces de usuários terem se tornado fáceis de usar e operar. Além disso, novas técnicas de IHC têm sido desenvolvidas, oferecendo um grande grau de liberdade se comparadas com as interfaces tradicionais (Rekimoto e Nagao, 1995).

De forma resumida, segue abaixo a evolução das interfaces desde a década de 50 (Valente, 2004):

1950: painéis com chaves seletoras - a interface se encontrava em nível de hardware e só era compreendida por engenheiros.

1970 - 1980: terminais - a interface aparece ao usuário como comandos de interação, que era realizada através de linha de comando.

1984 - 1986: GUI (*Graphic User Interface* - Interface Gráfica do Usuário) surgem às primeiras interfaces gráficas, iniciando a era das caixas de diálogos com o MacOS, sistema operacional desenvolvido pela empresa Apple Computer e o Microsoft Windows 1.0, desenvolvido pela Microsoft Corporation.

1990 - 2000: sistemas distribuídos - as interfaces passam de mono, para multiusuário, permitindo que equipes de usuários trabalhem cooperativamente, além disso, as interfaces ficam mais robustas e interativas. Ainda nestes anos surgem também a RVe RA como novos paradigmas de interface.

2.4. O Usuário e a Interface

Quando se define uma interface é necessário conhecer as características e necessidades do usuário que utilizará um determinado software. Neste tópico, é feito um estudo sobre o usuário e de aspectos que influenciam ou podem ajudar na definição da interface.

Antes de definir uma interface, é necessário o entendimento de algumas características dos usuários, tais como, que tipo de usuário utilizará o software e quais suas necessidades, bem como, a idade, educação, motivação e objetivos (Shneiderman, 1987).

Quando se cria uma interface, é necessário saber que cada usuário tem suas necessidades, limitações e preferências, isso torna o usuário uma abstração, cujas interações deverão ser projetadas visando alguns objetivos fundamentais.

Para isto, algumas características de usuários devem ser consideradas:

- O usuário nem sempre informa o que realmente necessita.
- Usuário deve ser envolvido no processo de criação da interface.
- Métodos desenvolvidos devem focar o uso do sistema por parte do usuário, em um contexto particular (Valente, 2004).

Os usuários trazem quase sempre, experiências passadas e geralmente já tem concepção daquilo que o computador pode fazer. Portanto, é necessário que a interface traga uma informação de forma perceptível, sem a necessidade de manuais ou treinamentos extensivos.

Esta informação perceptível pode ser feita, através de sinais emitidos pela interface ao usuário, como cor, textos, ícones, dentre outros. Deve-se evitar a frustração do usuário, para isso, é desejável que a informação chegue ao usuário de forma intuitiva e sem ambigüidades.

A habilidade humana de interpretação aliada aos recursos de interfaces, bem como, menus, imagens etc., possibilita a realização de tarefas de forma rápida e os computadores vão se tornando ferramentas, não apenas para profissionais especializados, mas a todas as classes de pessoas.

Um dos principais fatores humanos de interação com a máquina é o da comunicação visual, ou seja, é através da visão que se interpretam e se percebem as informações contidas na interface. O olho humano, juntamente com o cérebro, recebe e interpreta as informações visuais com base no tamanho, forma, cor e outras características (Pressman, 2006).

Todas estas informações associadas a uma interface, quando recebidas e interpretadas pelo usuário, irão dizer se esta interface trouxe ou não amigabilidade. Por isso, é importante o estudo destas informações, visando conhecer melhor o usuário, e consequentemente, atendendo melhor suas expectativas.

A Figura 2 ilustra alguns conceitos que devem fazer parte de um projeto de interface para o usuário.

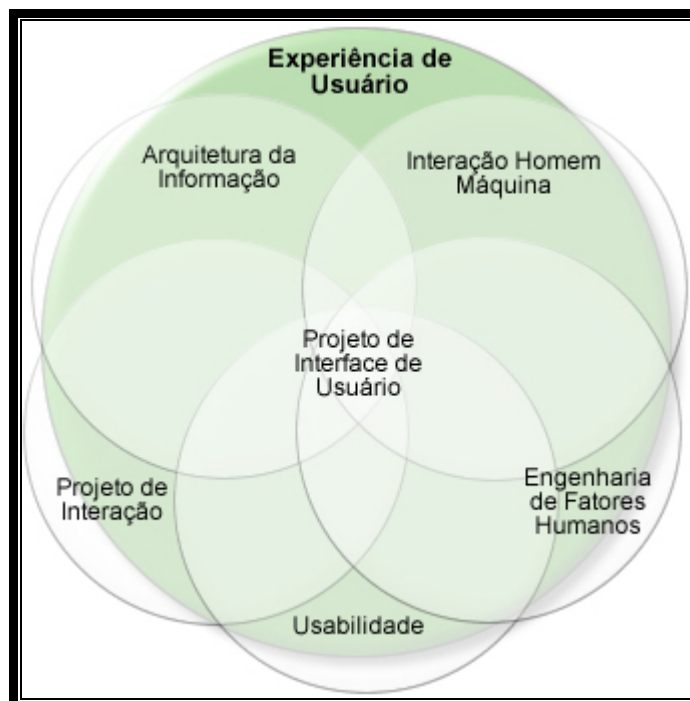


Figura 2. Diagrama relacionado com o projeto de interface de usuário

Fonte: (Paradyne, 2007)

Experiência de usuário: é um termo que descreve a experiência global e satisfação que um usuário tem ao usar um produto ou sistema. Para a construção de algum sistema deve-se

levar em conta a experiência do usuário e essa experiência envolve uma série de técnicas que ajudará na construção de uma interface, sendo: projeto de interação, arquitetura de informação, IHC, engenharia de fatores humanos, usabilidade e projeto de interface de usuário.

Projeto de interação: examina o comportamento inteligente do usuário em espaços físicos e virtuais. Quando o projeto de interação se trata das duas entidades humano e computador, o foco é criar um conjunto de tarefas que visa a observação do comportamento inteligente em realizar tais atividades.

Arquitetura da informação: é a arte de estruturar ou organizar o conhecimento. Um sistema será mais intuitivo e agradável ao usuário se a organização de informação for lógica e compreensível.

Usabilidade: é a mensuração da facilidade da utilização de uma ferramenta por uma pessoa para realizar uma atividade, além disso, geralmente é vista na engenharia de software e que está relacionada a aspectos de projetos de interface para usuário (Bevan apud Neves, 2005).

A usabilidade está associada a medidas e atributos que nos permite a avaliação e o desenvolvimento de sistemas e que tem um grande impacto na criação positiva da experiência do usuário.

A usabilidade é avaliada considerando se o uso do sistema é eficiente, eficaz e, sobretudo avalia também a satisfação do usuário.

Alguns atributos para medição de sistemas são definidos a seguir de acordo com a norma ISO 9241-11:

- **Eficácia** - o usuário rapidamente consegue explorar o sistema e realizar suas tarefas cotidianas com completude e acurácia;
- **Eficiência** - nesse atributo, é verificado se o usuário ao interagir com o sistema consegue realizar suas tarefas com alto nível de produtividade;
- **Facilidade** - após certo período sem utilizá-lo, o usuário não freqüente é capaz de retornar ao sistema e realizar suas tarefas sem a necessidade de reaprender como interagir com ele; e
- **Satisfação** - considera se o usuário encontra conforto com a utilização em um determinado produto;

A Figura 3 apresenta características de usabilidade. Conforme pode ser observado na Figura, foi marcado com um círculo azul e vermelho com setas, mostrando as diferentes possibilidades de acesso a uma mesma informação.



Figura 3. Exemplo de usabilidade na WWW

Fonte: (GLOBO, 2007)

Segundo o documento da ISO 9126 (1998), a usabilidade é conjunto de atributos importantes na avaliação individual de um sistema, assim, a usabilidade é vista como um dos atributos da qualidade de software. Além desses atributos definidos pelos padrões de qualidade, alguns estudiosos da área definem suas próprias regras e princípios de usabilidade. Segundo Constantine (1995), algumas regras fundamentam um bom projeto de interface:

A seguir são mostradas algumas dessas regras:

- **Acesso:** sistemas usáveis devem ser auto-explicativos para usuários que, embora não o conheçam, tenham experiência no domínio de aplicação a que o sistema se destina;
- **Eficácia:** sistemas usáveis não devem ter obstáculos para o uso eficiente por um usuário capacitado e com experiência no sistema;
- **Progressão:** sistemas usáveis possuem recursos adequados ao nível de experiência do usuário, facilitando o aumento progressivo de conhecimento e habilidades no uso do sistema;
- **Suporte:** sistemas usáveis dão suporte à atividade real que o usuário objetiva realizar de maneira, fácil, rápida e agradável; e

- Contexto: sistemas usáveis são adequados às condições e ambientes em que serão usados.

Interação humano-computador: é definido com o estudo de interação entre usuários e o computadores. A interação do usuário com um dispositivo físico é um exemplo prático de IHC e que não é uma prática comum da experiência do usuário em seu dia a dia.

Engenharia de fatores humanos: também conhecida com ergonomia, que tem como objetivo à concepção de instrumentos, máquinas e dispositivos que possam ser utilizados com o máximo de conforto, segurança e eficácia (Wisner, 1987). Ainda segundo (Wisner, 1987) a ergonomia de software é um caso particular de adaptação do trabalho ao homem - "a adaptação do sistema informatizado à inteligência humana".

User Interface Design: projeto de interface de usuário é o processo global de projetar a interação entre um humano (o usuário) e uma máquina (computador), incluindo a interface gráfica, informações e uma larga variedade de métodos de usabilidade. O projeto de interface fica no centro de todas as outras áreas citadas acima. Esta é a última meta para criar um mecanismo otimizado entre o usuário e o sistema.

2.5. Fundamentos de RV e RA

A Realidade Virtual é uma interface avançada para aplicações computacionais, onde o usuário pode navegar e interagir, em tempo real, em um ambiente tridimensional gerado por computador, usando dispositivos multisensoriais (Kirner, 2007).

Aplicações de RV podem ser desenvolvidas com as linguagens VRML, X3D, OpenGL e Java3D, OpenSceneGraph, entre outras.

O VRML (*Virtual Reality Modeling Language*) é uma linguagem que permite a criação de objetos virtuais principalmente para aplicações de RV na *World-Wide Web* (WWW), utilizando apenas um *browser* (navegador) e um *plug-in* para visualizar as aplicações. O *plug-in* é um programa que permite a visualização informações dentro da janela do *browser* da WWW.

OpenGL (*Open Graphics Library*) não é uma linguagem de programação, é uma biblioteca com uma sofisticada API (*Application Programming Interface*), para criação de aplicações gráficas 2D e 3D (OpenGL, 2007).

A API (*Application Programming Interface*) Java 3D consiste em uma hierarquia de classes Java que serve como interface para o desenvolvimento de sistemas gráficos tridimensionais (Sun Microsystems, 2007).

A Realidade Aumentada é a inserção de objetos virtuais no mundo real. A interface do usuário é aquela, que ele usa no ambiente real, adaptada para visualizar e manipular os objetos virtuais colocados no seu espaço. A Figura 4 ilustra alguns desses conceitos.

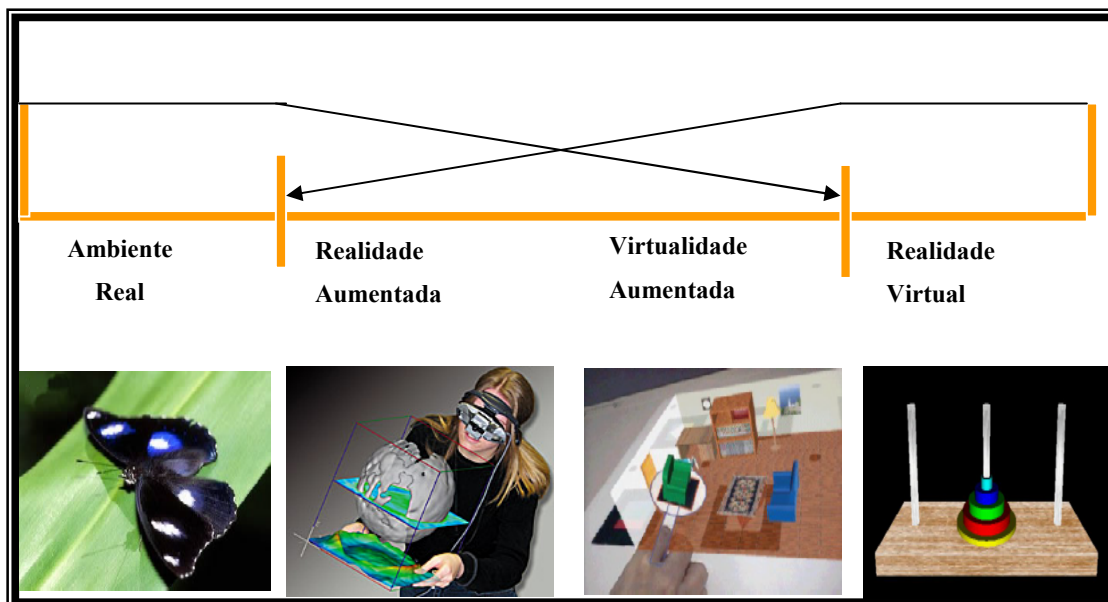


Figura 4. Continuum de Milgram

Fonte: (Milgram, 2006)

A Virtualidade Aumentada se caracteriza quando objetos virtuais são predominantes em um ambiente real.

Aplicações de RA podem ser desenvolvidas utilizando algumas bibliotecas tais como, OSGART, ARtag e ARToolKit.

O OSGART é uma biblioteca desenvolvida recentemente pelo *HITLabNZ* (*Human Interface Technology New Zealand*), que visa o desenvolvimento de aplicações de RA, utilizando as bibliotecas ARToolKit e a *OpenSceneGraph*. Essa última biblioteca permite alta qualidade de renderização, além da possibilidade de inserção de vídeos no ambiente real, mais informações podem ser observadas em (OSGART, 2007).

O ARtag é um sistema que permite a inserção de objetos virtuais e vídeos no ambiente real. Além disso, esse sistema ao invés de usar um único marcador, utiliza um conjunto de marcadores para exibir um objeto virtual, mais informações podem ser encontradas em (ARtag, 2007).

O ARToolKit é uma biblioteca gratuita criada na Universidade de Osaka no Japão, pelo Dr. Hirokazu Kato. Atualmente a biblioteca é desenvolvida pelo Laboratório de Tecnologias de Interface Humana, da Universidade de Washington (EUA) e da Universidade de Canterbury (Nova Zelândia). Por ser de código aberto, permite que programadores

desenvolvam ainda mais a biblioteca, a fim de contribuir para o desenvolvimento de aplicações em RA (Kato, Billinghurst et al., 2005).

Uma das maiores dificuldades no desenvolvimento de aplicações de RA é o cálculo preciso do ponto de vista do usuário em tempo-real para que objetos virtuais estejam exatamente alinhados com cenas do mundo real. O ARToolKit resolve o problema utilizando técnicas de visão computacional para calcular a posição real da câmera e orientação relativa aos marcadores, permitindo ao programador sobrepor objetos virtuais sobre estes marcadores (Billinghurst, 2005).

O ARToolKit baseia-se na utilização de cartões marcadores, esses marcadores são símbolos que permitem o rápido conhecimento da posição e orientação da câmera, facilitando o cálculo em tempo real da movimentação necessária para a sobreposição dos objetos virtuais.

Atualmente, existem várias versões do ARToolKit implementadas em várias linguagens, como por exemplo em Matlab e Java. Mas as versões mais utilizadas e atualizadas do software são as versões desenvolvidas na linguagem de programação C++. Algumas dessas versões possuem suporte a VRML, mas a maioria está disponível apenas para OpenGL.

Os principais requisitos de hardware para desenvolver e executar aplicações do ARToolKit são uma câmera digital, Sistemas Operacionais Windows(95/98/2000/XP) ou Linux, marcadores e um computador igual ou superior a 500 Mhz (Kato, Billinghurst et al., 2000).

A linguagem C++, além do poder da orientação a objetos, possui ainda flexibilidade no manuseio de memória, bem como um alto desempenho nas aplicações desenvolvidas. Além disso, a linguagem C++ pode ser utilizado em diversas plataformas devido sua portabilidade de código. Alguns detalhes de funcionamento do ARToolKit serão mostrados a seguir.

Através de uma câmera de vídeo, uma imagem é capturada (Figura 5(a)) e transformada em uma imagem binária (Preto e Branco) baseado em um limiar de intensidade (Figura 5 (b)).

Depois, busca-se nesta imagem por regiões quadradas. O ARToolKit encontra todos os quadrados na imagem binária, muitos deles não correspondem a marcadores de referência.

Para cada quadrado, o desenho padrão dentro dele é capturado e comparado com alguns marcadores pré-treinados.

Se houver alguma similaridade, o ARToolKit considera que encontrou um dos marcadores de referência.

O ARToolKit usa então o tamanho conhecido do quadrado e a orientação do padrão encontrado para calcular a posição real da câmera em relação a posição real do marcador. Uma matriz 3x4 conterá as coordenadas reais da câmera em relação ao marcador. Esta matriz é usada para calcular a posição das coordenadas da câmera virtual. Se as coordenadas virtuais e reais da câmera forem as mesmas, o objeto virtual pode ser desenhado precisamente sobre o marcador real (Figura 5 (c)).

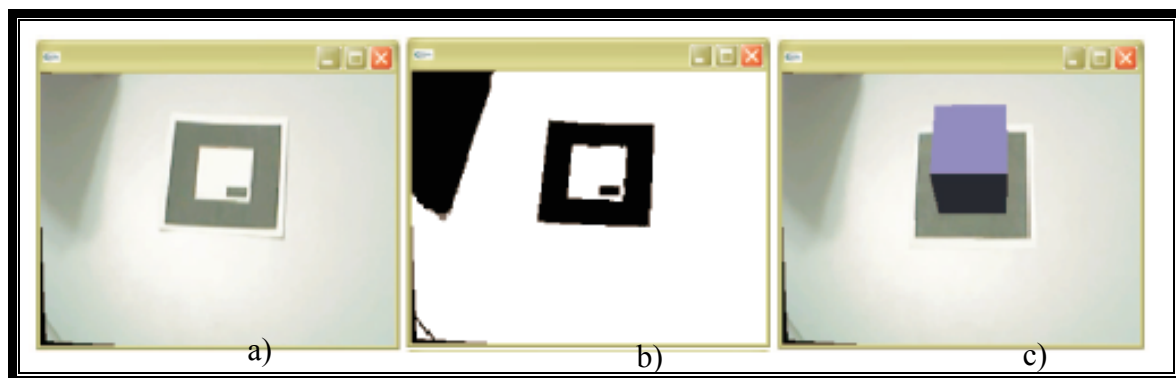


Figura 5. Análise da imagem pelo ARToolKit.

Fonte: (Kato, Billinghurst et al., 2005)

Como já foi visto o ARToolKit usa técnicas de Visão Computacional para calcular o ponto de vista real da câmera em relação a um marcador no mundo real. A Figura 6 mostra os passos do processo de funcionamento do ARToolKit.



Figura 6. Processamento de imagens do ARToolKit.

Fonte: (Kato, Billinghurst et al., 2005)

2.6. Etapas de utilização do ARToolKit

Nesta seção serão mostradas todas as etapas que um usuário precisa realizar para executar uma aplicação no ARToolKit. A Figura 7 mostra como é a estrutura de diretório do ARToolKit para facilitar o entendimento dos próximos passos.

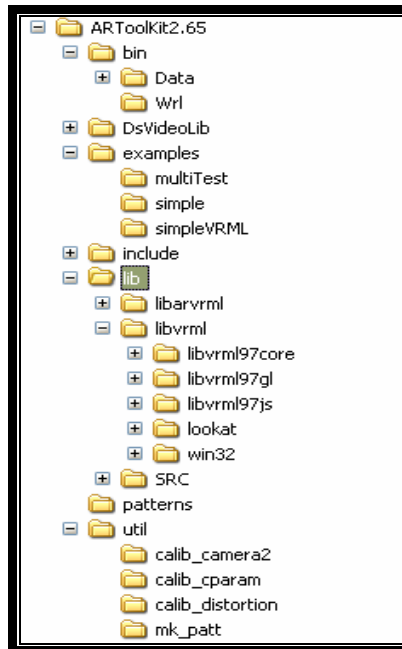


Figura 7. Estrutura ARToolKit.

O objetivo não é detalhar cada diretório mostrado na Figura 7, mas dar uma visão geral para melhor entender as partes necessárias para a execução deste software. O diretório **bin** contém todos os programas do ARToolKit que podem ser executados, tais como programas de calibração de câmera, de cadastramento de marcadores, dentre outros. Os códigos fonte destes programas ficam no diretório **examples**.

Dentro do diretório **bin** contém dois subdiretórios: **Wrl** e **Data**, onde ficam armazenados todos objetos virtuais em VRML, juntamente com arquivos no formato **.dat**, e todos os marcadores respectivamente. Os arquivos **.dat** serão vistos posteriormente.

As bibliotecas do ARToolKit ficam no diretório **lib** e o código fonte completo para as bibliotecas, no diretório **lib/SRC**. A seguir, serão mostrados alguns passos necessários para o desenvolvimento de uma aplicação em RA.

Passo 1: Primeiro é necessário que o usuário copie seu objeto virtual em VRML com todas as texturas para dentro do diretório **Wrl** do ARToolKit.

Passo 2: No segundo passo, deve-se criar um arquivo com extensão **.dat**, por exemplo, **bee.dat**. A Figura 8 mostra um exemplo de conteúdo desse arquivo.

Wrl/bud_B.wrl	#nome do arquivo .wrl a ser alterado
0.0 0.0 0.0	#translação
0.0 0.0 0.0 0.0	#rotação
2.5 2.5 2.5	#escala

Figura 8. Conteúdo do arquivo .dat

O arquivo **bee.dat** mostrado acima, deve ser associado com um outro arquivo **.wrl**, isso pode ser feito editando e alterando o nome da primeira linha do arquivo **beer.dat**, adicionando um outro nome de um arquivo **.wrl** que esteja no diretório Wrl, conforme visto no passo 1.

Passo 3: Nesse passo, é necessário criar um marcador, esse marcador deve aproximadamente 80mm de largura e altura, a Figura 9 mostra o exemplo de um marcador.



Figura 9. Exemplo de marcador.

Fonte: (Kato, Billinghamurst et al., 2005)

Depois de criado um novo marcador, é necessário cadastrá-lo no ARToolKit, para isso, utiliza-se um aplicativo chamado Mk_Patt. Esse aplicativo fica localizado dentro da pasta **bin** do ARToolKit. A Figura 10 mostra a execução desse aplicativo para o cadastramento do marcador mostrado anteriormente.

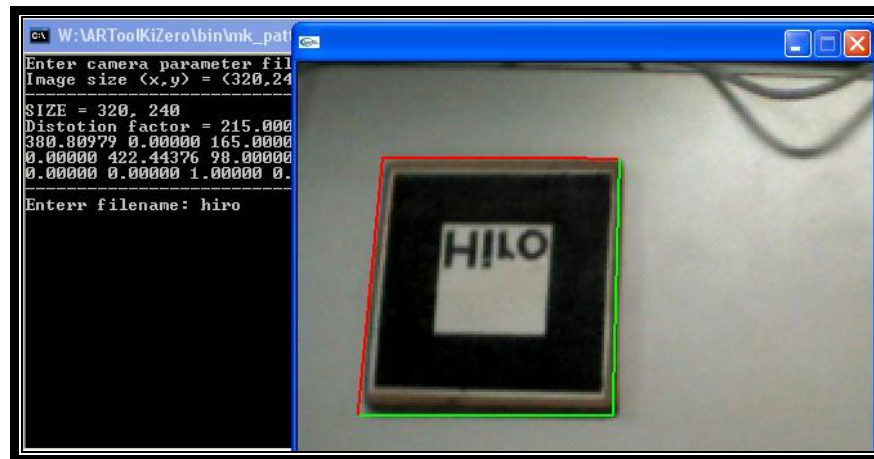


Figura 10. Cadastramento de marcador.

A Figura 10 mostra o aplicativo Mk_Patt em processo de execução. A câmera deve ser posicionada em cima do marcador. A linha vermelha deve estar posicionada do lado esquerdo e a linha verde no lado direito do marcador. Isto significa que o ARToolKit encontrou um determinado marcador. Após esse processo, o usuário clica com o botão esquerdo do mouse na imagem e uma tela no formato DOS é exibida para que se entre com o nome do marcador, que no nosso caso foi chamado de hiro, conforme foi exibido na Figura 10.

Quando um novo marcador é cadastrado, o arquivo é armazenado fora do diretório correto. Assim é necessário que o usuário copie esse arquivo para um diretório no ARToolKit chamado data. O próximo passo agora é editar um arquivo chamado vrml_data, esse arquivo contém um lista de marcadores associados com arquivos .dat criado no passo 2. A Figura 11 mostra um exemplo clássico do arquivo vrml_data (Kato, Billinghurst et al., 2005).

```

1: #the number of patterns to be recognized
2: 1
3: #pattern 1
4: VRMLWrl/snoman.dat
5: Data/Y
6: 80.0
7: 0.0 0.0

```

Figura 11. Conteúdo do arquivo vrml_data

A linha 2 do arquivo vrml_data mostrado no trecho de código da Figura 11, é referente ao número de marcadores que serão carregados pelo ARToolKit. A linha 4, informa o nome do arquivo .dat, que por sua vez, está ligado com o nome do arquivo .wrl. A linha 5 contém o nome do marcador.

Para finalizar o processo de cadastramento de marcadores, um exemplo será mostrado na Figura 12, com o arquivo vrml_data alterado para conter o marcador da Figura 9 visto

anteriormente, esse é o último passo que deve ser feito pelo usuário, antes de executar o ARToolKit para exibir suas aplicações.

```
1: #the number of patterns to be recognized
2: 2
3: #pattern 1
4: VRMLWrl/snoman.dat
5: Data/Y
6: 80.0
7: 0.0 0.0
8: #pattern 2
9: VRMLWrl/bee.dat
10: Data/hiro
11: 80.0
12: 0.0 0.0
```

Figura 12. Arquivo vrml_data alterado

O trecho de código da Figura 12, apresenta o acréscimo de um novo padrão (#pattern2), linha 8, na linha 9 o usuário deve escrever o nome do arquivo *.dat*, que no caso presente é o *bee.dat*. Na linha 10, adiciona-se o nome do marcador cadastrado, por fim, a linha 2 deve ser alterada para que os dois marcadores sejam carregados pelo ARToolKit.

Cada padrão (marcador) é associado com um objeto virtual, como já foi visto, esta associação é feita através do arquivo vrml_data. A Figura 13 mostra um diagrama com a estrutura das associações vistas nos passos anteriores.

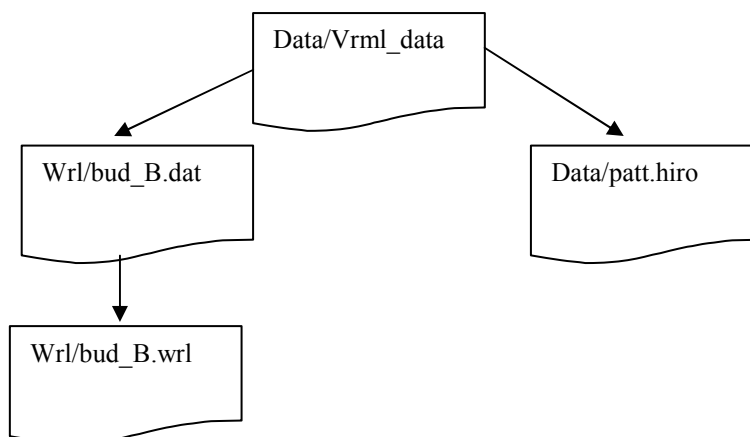


Figura 13. Diagrama de associações.

No topo do diagrama, o arquivo vrml_data contém as seguintes informações: padrões (patt_hiro), parâmetros de transformações geométricas (bud_B.dat) e objetos virtuais (bud_B.wrl).

2.7. Considerações Finais

Este Capítulo abordou teorias de IHC, que podem auxiliar no desenvolvimento de interfaces mais interativas e objetivas. Além disso, foram estudados os fundamentos da biblioteca ARToolKit, onde é mostrado passo a passo, as etapas que um usuário comum de computador deve fazer para se utilizar dessa biblioteca. Na Figura 10, percebe-se que a interface utilizada para inserir um novo marcador no sistema, ainda é feita ainda na forma de interface DOS (*Operating system of Directory*). Além disso, foi visto também nos passos necessários para a criação de uma aplicação de RA, que o usuário precisa fazer edição de arquivos e cópias para diretórios específicos. Isso tudo, mostra que para se utilizar o ARToolKit é necessário executar uma série de passos não triviais. No próximo Capítulo, baseado nos fundamentos aqui estudados, é apresentado a especificação do sistema, bem como sua arquitetura e as tecnologias de apoio, no qual permitem uma melhor usabilidade do sistema desenvolvido.

Capítulo 3

3. TRABALHOS RELACIONADOS

3.1. Introdução

Este Capítulo tem como objetivo a avaliação de alguns softwares de RA. Foram analisados alguns trabalhos relacionados, que utilizam a biblioteca ARToolKit buscando verificar a existência de alguma interface que facilite a manipulação dessa biblioteca por usuários comuns.

3.2. AR System

Desenvolvido pela *University of Aizu*, o AR System (Tsuchimoto *et al*, 2004) é uma implementação de uma interface de Realidade Aumentada para se fazer medidas de regiões do corpo humano. Usando um sensor tátil o sistema provê medidas em tempo real dessas regiões. Para a utilização desse sistema utiliza-se um HMD (*Head Monted Display*) com uma câmera de vídeo e um computador. A Figura 14 ilustra como é um sistema de RA com HMD.

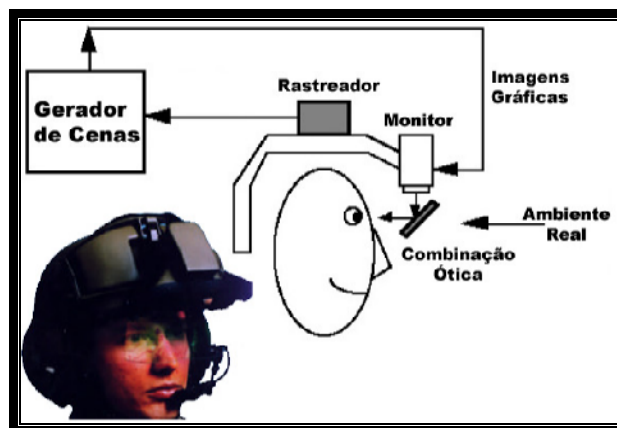


Figura 14. Sistema de RA.

Fonte: (Azuma, 1997)

Este dispositivo compõe-se de óculos semitransparentes ou transparentes que além de exibir as imagens virtuais, permite que o utilizador veja também o mundo real. Além disso é composto por duas partes: um gerador de cenas, que recebe os dados e faz os cálculos necessários, gerando imagens virtuais, e os monitores dos óculos que, por sua vez, reproduzem as imagens virtuais sobrepostas ao mundo real visto através das lentes do HMD.

As tecnologias envolvidas no desenvolvimento do AR System foram: ARToolKit, DirectX,(Microsoft, 2007) uma biblioteca que captura quadros de vídeo da câmera; OpenGL (OpenGL,2007) uma biblioteca para modelagem e renderização de objetos em duas e três dimensões; por fim, a FLTK (Fast Light Tool Kit) uma biblioteca para a criação de interfaces gráficas para a linguagem C++ (Spitzak, 1991).

O rastreamento do objeto real é feito utilizando um marcador fixado ao corpo do paciente. Os resultados são exibidos em três dimensões em cima do corpo do paciente.

O sistema desenvolvido pode ser utilizado por médicos em exames físicos envolvendo membros do corpo humano.

A interface gráfica do usuário foi criada para que o médico tenha facilidade ao visualizar resultados de exames dos pacientes, a Figura 15 mostra interface do software AR System.

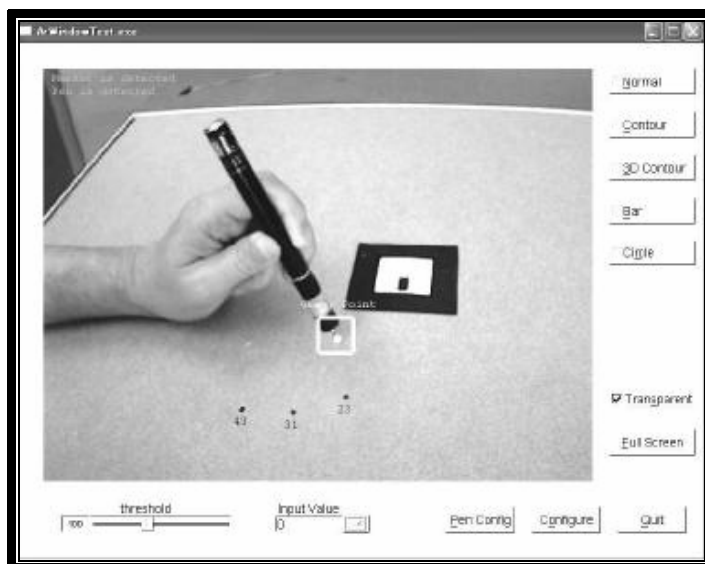


Figura 15. Interface AR System

Fonte: (Tsuchimoto et al, 2004)

O AR System conta com cinco técnicas de visualização, cada uma delas são descritas abaixo (Tsuchimoto *et al*, 2004).

1 - Normal: Os dados são exibidos em forma de pontos e números, que representam a localização das medidas e seus valores.

2 - Círculos: Os dados exibidos em círculos e em cores diferentes mostram valores pequenos e grandes para a localização das medidas dos membros no paciente.

3 - Barras: Os resultados das medidas são mostrados em barras de tamanhos variados para distinguir valores de medidas.

4 - Contornos em 2D e 3D: Servem para destacar os exames em forma de contorno coloridos.

O sistema apresenta várias formas de visualização dos dados. No entanto, suas funcionalidades são específicas para medição de características de tecidos do corpo humano. Portanto, o sistema oferece uma interface muito específica beneficiando os profissionais da área médica, mas não permite a criação de aplicações de RA por usuários potenciais. O sistema não apresenta uma alternativa para a inserção de novos marcadores e não viabiliza uma interface que realize associações de marcadores com objetos virtuais.

3.3. AR Pad

O AR Pad é uma interface que permite a interação colaborativa de usuários em um ambiente de Realidade Aumentada. Para isso, utilizam um handheld display juntamente com uma câmera e um SpaceBall. Assim, o AR Pad explora como um handheld display pode ser usado para visualizar e interagir com ambientes colaborativos em RA (Mogilev et al, 2002).

O AR Pad consiste de três componentes principais: um dispositivo de entrada chamado SpaceBall conectado em um handheld LCD(Liquid Crystal Display), e uma câmera. Além disso, um computador é necessário para a geração dos ambientes virtuais. A Figura 16 e 17 mostram alguns desses periféricos e funcionamento desse sistema respectivamente.



Figura 16. Spaceball e handheld

Fonte: (Mogilev et al, 2002)

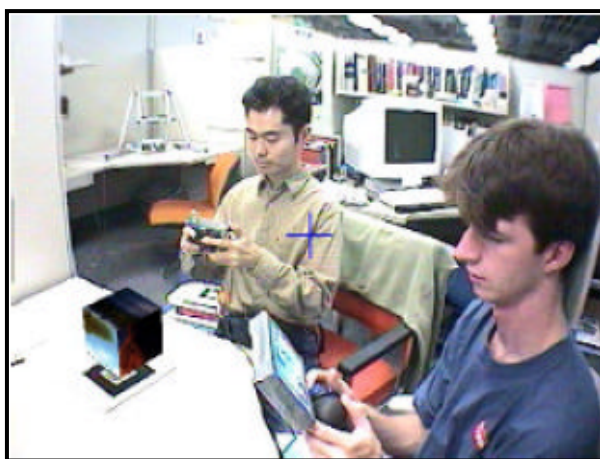


Figura 17. Ambiente Colaborativo

Fonte: (Mogilev et al, 2002)

Um ambiente colaborativo permite que alterações sejam feitas por dois ou mais usuários. A imagem capturada pela câmera é transmitida para um computador e retornada para um *handheld LCD*. Como o ARToolKit possui um poder de processamento e cálculo em

tempo real da posição da câmera em relação ao marcador, ele foi usado para processar a imagem do marcador capturado pela câmera e projetar o objeto virtual no mundo real.

Uma vez que o objeto virtual já esteja posicionado em cima do marcador, ele pode ser selecionado através do *SpaceBall* e movido em diversas posições na tela do *handheld*. Além disso, o AR Pad pode ser usado em um ambiente colaborativo, onde vários usuários com um *handheld* compartilham um mesmo marcador, isso implica que, cada usuário pode manipular e interagir um mesmo objeto virtual.

O sistema AR Pad apresenta características importantes, tais como manipulação e interação, e capacidade de suporte a vários usuários em um ambiente colaborativo. O sistema apresenta uma interface para manipulação de objetos virtuais no mundo real, além disso, é um sistema colaborativo. Porém, não aborda como adicionar um objeto virtual específico ou até mesmo a inserção de novos marcadores no sistema, e nem mesmo a associação de objetos virtuais a marcadores.

3.4. Tiles System

O Tiles System é um sistema baseado na biblioteca ARToolKit, que provê uma interface genérica para manipulação e interação em ambientes de RA.

O objetivo desse sistema é criar um ambiente onde o usuário tem condições de interagir com objetos virtuais de forma dinâmica, assim como se faz no mundo real.

Para fazer uso desse sistema o usuário precisa, além de um computador e marcadores, necessita também de um HMD acoplado com uma câmera e um quadro branco que serve de superfície para apoio dos marcadores (Poupyrev et al., 2002). A Figura 18 mostra o ambiente de trabalho do usuário.

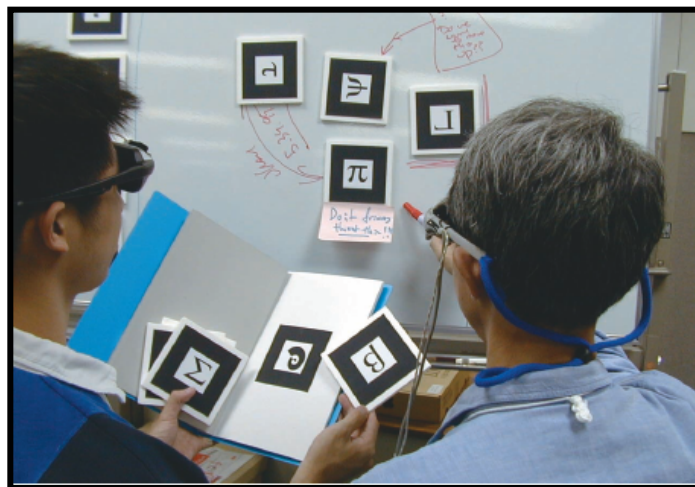


Figura 18. Usuários manipulando dados em um ambiente colaborativo do Tiles System

Fonte: (Poupyrev et al, 2002).

Para introduzir o usuário no sistema, proporcionando mais interação e manipulação de objetos virtuais, o Tiles System conta com três marcadores de controle. Cada um desses marcadores oferece uma operação básica. Através dessas operações o usuário pode copiar, apagar ou obter ajuda no ambiente, através do marcador *help*. Com esses marcadores de operações o usuário pode, por exemplo, copiar um elemento virtual de um marcador para outro, além disso, o sistema permite a participação mútua e colaborativa entre os usuários. As Figuras 19 e 20 mostram os três marcadores de controle disponíveis para o usuário e operações de remoção de objetos virtuais respectivamente.

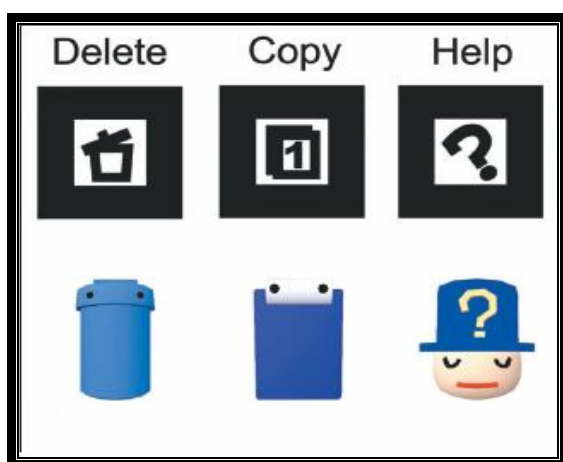


Figura 19. Marcadores de Controle

Fonte: (Poupyrev et al, 2002).



Figura 20. Utilização do marcador delete

Fonte: (Poupyrev et al, 2002).

Pode se perceber que o Tile System apresenta uma alternativa viável no que diz respeito à interação em ambientes de RA. O sistema também faz associações de objetos virtuais aos marcadores por meio do marcador de controle denominado *Copy*. No entanto, o sistema não faz menção de um mecanismo para a inserção de novos marcadores no ambiente, o que facilitaria o uso do sistema por usuários leigos no assunto.

3.5. Considerações Finais

Analisando os trabalhos relacionados pode se perceber o uso do ARToolKit em muitas aplicações de RA. Algumas características importantes foram destacadas, tais como, objetivos e usabilidade dos sistemas. Por exemplo, o primeiro sistema analisado, o AR system apresenta uma interface bastante intuitiva, com opções variadas de visualização dos dados, contudo, é um sistema que não apresenta nenhum recurso para inserção de novos marcadores.

O AR Pad e o Tiles System têm características similares, tais como, uma interface não convencional, e ainda permite manipulação e interação colaborativa entre os usuários. O Tiles

System apresenta ícones em forma de marcadores que dá ao usuário a possibilidade de realizar três operações, sendo uma delas, a associação de objetos virtuais a marcadores. Assim como o primeiro sistema analisado, as interfaces do Tiles System, e do AR Pad não permitem a inserção de novos marcadores no sistema.

Tomando como referencia os trabalhos analisados, pode se perceber a necessidade de se desenvolver uma interface intuitiva para facilitar a adição e associação de novos marcadores, facilitando a utilização do ARToolKit por usuários potenciais.

Capítulo 4

4. ESPECIFICAÇÃO DO SISTEMA PROPOSTO

4.1. Introdução

A partir dos trabalhos relacionados estudados no Capítulo 3, é proposta uma arquitetura de um sistema que permite, por meio de uma interface gráfica, a execução de todos os passos mostrados no item 2.6 do Capítulo 2, tais como, criação, cadastramento e associação de marcadores, de forma rápida e mais intuitiva.

4.2. Tecnologias de Apoio

Nesta seção, são apresentadas as tecnologias de apoio que foram selecionadas para o desenvolvimento do sistema. Posteriormente será mostrada a integração dessas tecnologias para suportar a arquitetura proposta.

Para a implementação do projeto, foi escolhido o ambiente de programação, Borland C++ *Builder 6.0*, por ser uma plataforma que apresenta uma grande variedade de componentes, tornando mais adequada para o desenvolvimento das interfaces gráficas e dos módulos de gerenciamento, que serão descritos mais adiante. Além disso, foi utilizado o *Microsoft Visual C++ 7.0* e a biblioteca GLUI para a construção de um dos módulos, essas ferramentas foram utilizadas por ser nativas do software ARToolKit.

4.3. GLUT e GLUI

A biblioteca GLUT (*OpenGL Utility Library*) possibilita a modelagem de várias primitivas geométricas, tais como, esferas, cones, torus etc (GLUT, 2007).

A GLUI (*Graphic Library User Interface*) é uma biblioteca em linguagem C++, baseada na biblioteca GLUT, provê ao usuário controles bem como, botões, caixa de textos e funções específicas que operam em conjunto para o desenvolvimento de aplicações em OpenGL. Além disso, esta biblioteca permite a criação de interfaces independente de plataforma.

As duas bibliotecas trabalham em conjunto, a GLUI serve para criação de interfaces e a GLUT é responsável por um conjunto de primitivas que são usados para a modelagem de objetos. A Figura 21 mostra um exemplo de modelagem com o *teapot* (pote de chá) e a interface com o painel de controle da biblioteca GLUI ao lado.

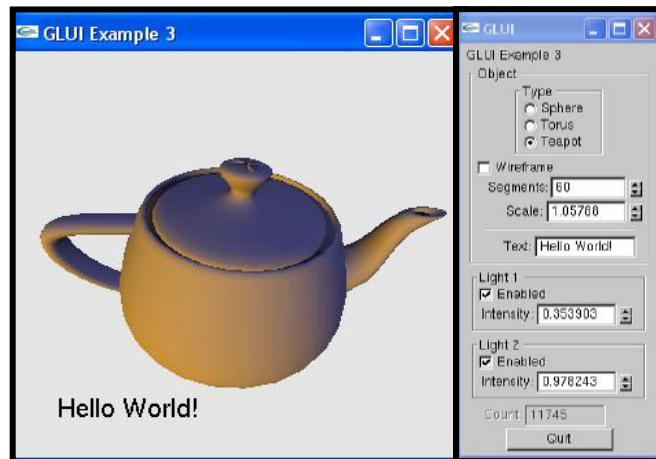


Figura 21. Exemplo de objeto virtual e painel de controle

Fonte: (Rademacher, 2005)

4.4. Arquitetura de Módulos do Sistema

A Figura 22 apresenta uma visão geral da arquitetura do sistema, composta pelos seguintes módulos: GUI, Criador de Marcador, Cadastrador de Marcador, Associador de Marcador e Executor RA.

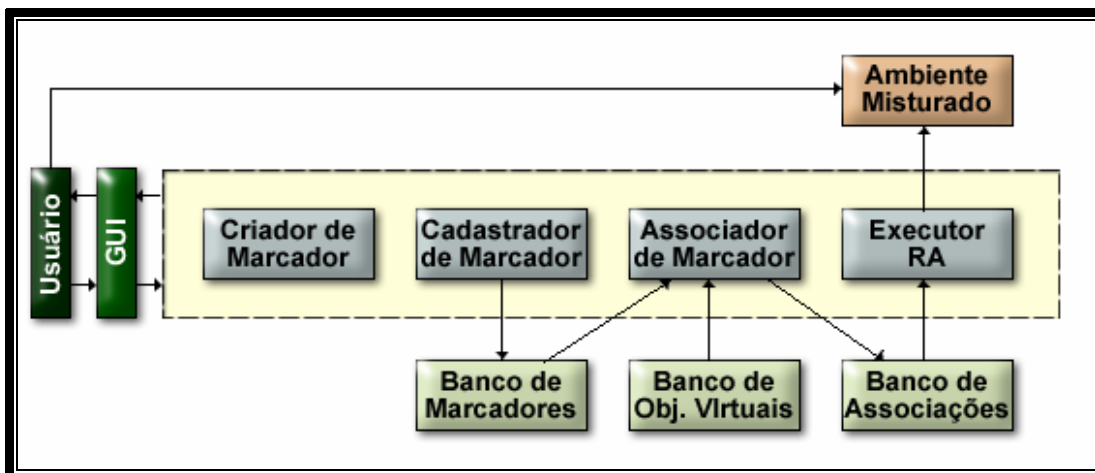


Figura 22. Arquitetura do Sistema.

O módulo **Criador de Marcador** permite que o usuário crie um determinado marcador. O módulo **Cadastrador de Marcador** gera um arquivo que contém informações de um determinado marcador, que é direcionado para o **Banco de Marcadores**. O módulo **Associador de Marcador** utiliza o **Banco de Marcadores** e o **Banco de Obj. Virtuais**, com isso, é gerado um novo **Banco de Associações**, que por sua vez é utilizado pelo módulo **Executor RA** para gerar o **Ambiente Misturado** onde o usuário visualiza.

4.5. GUI – Interface Gráfica

O módulo GUI - Interface Gráfica do Usuário foi uma nova camada desenvolvida no ambiente *Borland C++ Builder* e GLUI e que possibilita a comunicação do usuário com todos os outros módulos desenvolvidos. A utilização de componentes, tais como menus, botões, caixa de texto, permitem aos usuários a criação, inserção de marcadores e associação entre marcadores e objetos virtuais em VRML, possibilitando o desenvolvimento de aplicações em RA de forma mais rápida e intuitiva. A Figura 23 apresenta a interface inicial do sistema.

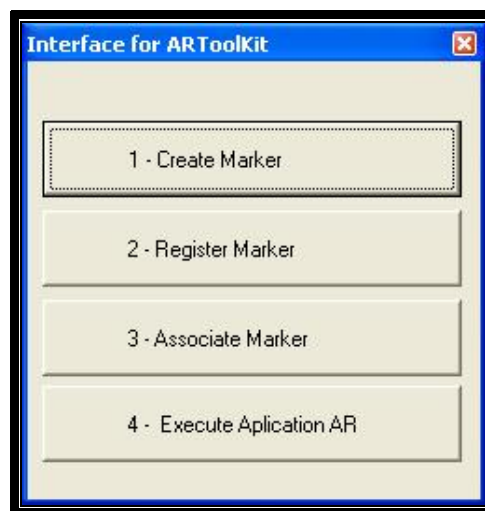


Figura 23. Tela inicial do Sistema

Como mostrado na Figura 23, a tela inicial do sistema apresenta quatro botões de acesso, que possibilitam ao usuário realizar passo a passo as etapas, que serão explicadas nos tópicos a seguir.

4.6. Módulo Create Marker

Através do ambiente de desenvolvimento *Borland C++ Builder*, foi possível criar a classe Marcador, que permite ao usuário à concepção de novos marcadores. Para que fosse possível gerar novos marcadores, carrega-se para um formulário no sistema, uma imagem binária (preto e branco) de 80mm. Através de uma caixa de ferramentas, tais como, desenho de um retângulo, desenho de elipse, borracha e pincel, o usuário pode criar seu próprio marcador. A Figura 24 mostra a tela responsável por essa tarefa.

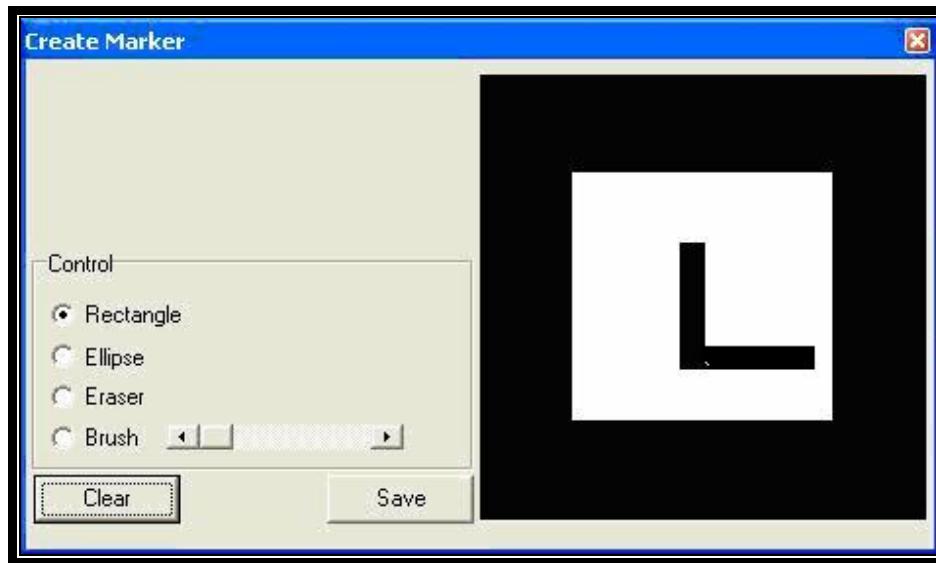


Figura 24. Interface Create Marker.

4.7. Módulo Register Marker

Este módulo permite à inserção de novos marcadores no sistema, de forma intuitiva e mais rápida. O ARToolKit já possui este módulo, no entanto, a inserção de novos marcadores pelo método tradicional do ARToolKit é feito com o uso de um aplicativo chamado Mk_Patt. Esse aplicativo já foi mencionado na Seção 2.6 passo 3 do Capítulo 2, onde foi exposto a forma de se cadastrar um novo marcador.

Para a implementação deste módulo, foi necessário reconstruir parte do aplicativo Mk_Patt. Para isso, foi utilizada a biblioteca GLUI. Essa biblioteca possui um conjunto de rotinas que permite a criação de interfaces em um ambiente de desenvolvimento usando a linguagem C++. Além disso, foi utilizado o *Microsoft Visual C++*, visto que, as bibliotecas do ARToolKit foram todas construídas usando este ambiente, inclusive o aplicativo Mk_Patt.

O *Microsoft Visual C++* permitiu a integração da biblioteca GLUI e do programa Mk_Patt. Com isso, foi construída uma interface que possibilita ao usuário entrar com o nome do marcador. Depois da inserção do nome do marcador e pressionado o botão OK, automaticamente o arquivo de matrizes contendo o marcador, é armazenado em um diretório específico, no qual será utilizado pelo próximo módulo da interface, conforme apresentado na próxima seção. A grande vantagem da implementação desse módulo em relação ao tradicional uso do aplicativo Mk_Patt, é que essa interface utiliza mecanismos intuitivos para o cadastramento de novos marcadores. A interface desse módulo é apresentada na Figura 25.

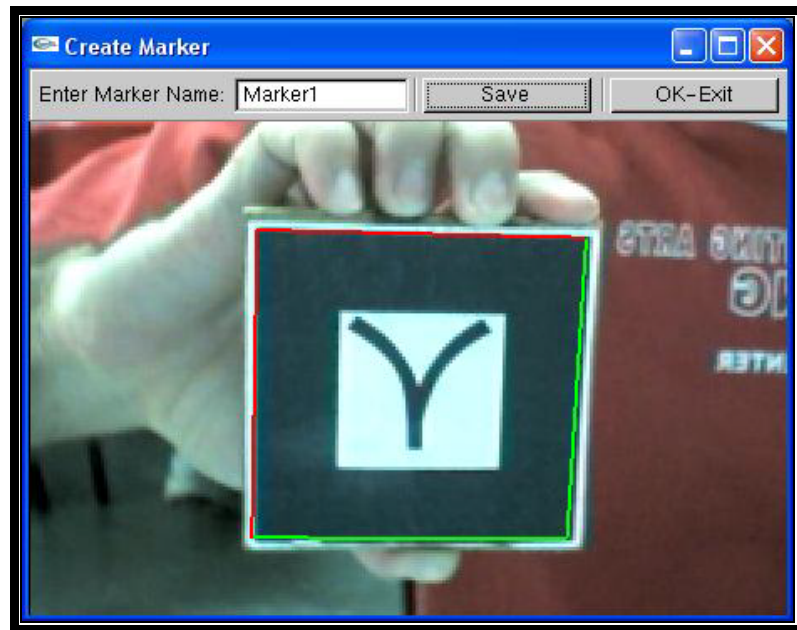


Figura 25. Interface Register Marker.

4.8. Módulo Associate Marker

Para se exibir um objeto virtual no ARToolkit são necessários alguns passos, tais como, gerar um novo marcador, cadastrar esse novo marcador no sistema e por fim, fazer a associação dos marcadores com objetos virtuais.

Associar marcadores a objetos virtuais no ARToolkit não é uma tarefa trivial para usuários comuns de computadores, conforme foi visto na Seção 2.6. Por isso, para que o ARToolkit possa ser utilizado por usuários não especializados, foi criado um módulo de interface que permite a realização dessa tarefa de forma visual, assim como nos outros módulos.

A interface apresenta dois *ListBoxs* (Lista), exibindo todos os marcadores cadastrados pelo módulo anterior e todos os objetos virtuais em VRML do usuário. Assim, o usuário pode escolher qual objeto virtual quer associar com um determinado marcador, além do nome dos objetos que são apresentados numa lista, ainda é possível visualizar através de imagens os objetos virtuais e os marcadores, dessa forma, fica ainda mais intuitivo fazer esta associação.

Assim que são feitas as associações, é gerado um arquivo contendo todos os parâmetros necessários para a utilização do sistema. A Figura 26 mostra esta interface.

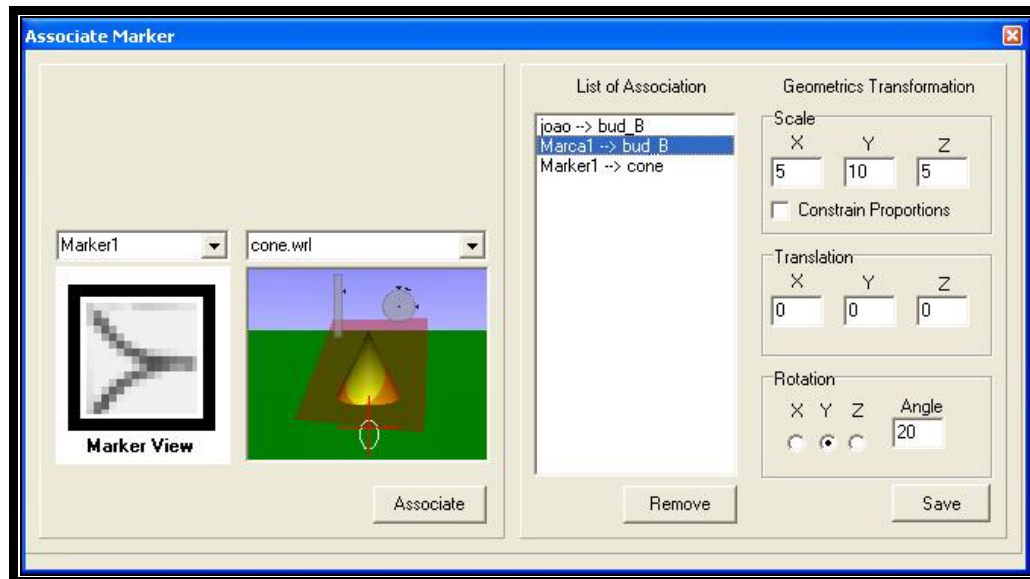


Figura 26. Tela responsável pelas associações dos marcadores.

Através do botão *Execute Application AR*, o usuário pode executar o aplicativo principal do ARToolKit e assim exibir suas aplicações (criadas nos módulos anteriores). Para exemplificar essa última etapa, a Figura 27, mostra uma aplicação de sólidos geométricos.



Figura 27. Tela de exibição de aplicações.

4.9. Especificação do Sistema

Para a fase de especificação do sistema foram utilizados os diagramas de caso de uso e diagramas de classes. Para isso, foi utilizada a linguagem UML (*Unified Modelling Language*), uma linguagem gráfica para visualização, especificação e documentação dos

artefatos que façam uso de sistemas de software (Microsystems, 2001). O modelo de caso de uso descreve a funcionalidade do sistema, o diagrama de classes é construído a partir de vértices e arcos que mostram um conjunto de classes, interfaces e colaborações e seus relacionamentos.

4.10. Diagrama de Casos de Usos

Os diagramas de casos de usos servem para especificar o comportamento de um sistema ou parte de um sistema, além disso, descreve uma seqüência de ações incluindo variantes realizadas pelo sistema para produzir um resultado observável do valor de um ator. Em seguida são descritos os casos de uso para o ator usuário comum, conforme mostrado na Figura 28.

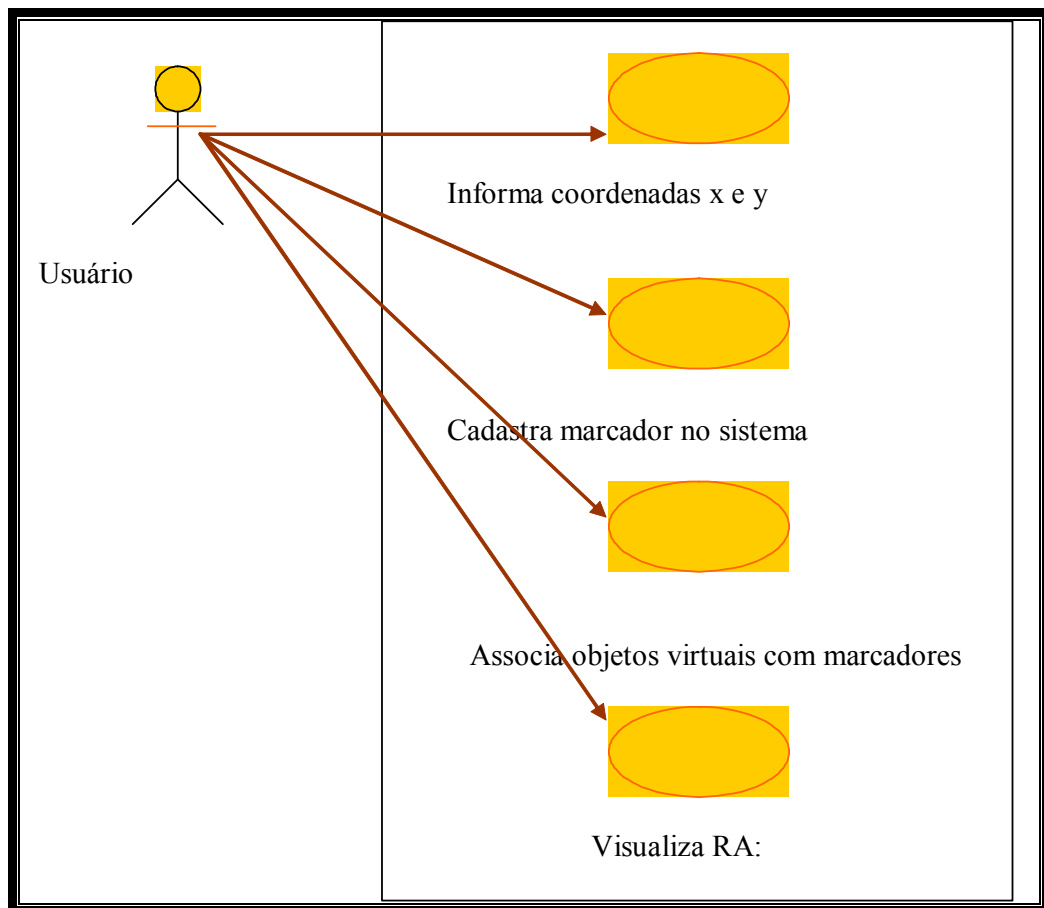


Figura 28. Diagrama de casos de usos.

Informa coordenadas x e y: o usuário informa às coordenadas x1 e y1 (linha e coluna), x2 e y2 (linha e coluna), além de informar também as ferramentas, retângulo ou elipse para que um novo marcador seja desenhado.

Cadastra marcador no sistema: o usuário solicita através de um botão o cadastramento de um novo marcador, para isso, o usuário exibe para a câmera o novo marcador criado na ação anterior. Tendo feito isso, o sistema gera um arquivo de matrizes contendo as informações de cada marcador usado para a associação na etapa seguinte.

Associa objetos virtuais com marcadores: nesta etapa o usuário pode associar objetos virtuais previamente inseridos no sistema, com os novos marcadores cadastrados.

Visualiza RA: o usuário solicita através do quarto botão da interface principal, a execução do pacote ARToolKit, no qual permite a visualização de objetos virtuais sobre marcadores reais. Para isso, o ARToolKit faz a leitura de um arquivo contendo nomes de marcadores e objetos virtuais, que foram associados nas etapas anteriores.

4.11.Diagrama de Classes

O diagrama de classes é uma ferramenta utilizada para a fase de análise e projeto do sistema. Esse diagrama contém as classes do sistema proposto e mostra a interação entre as classes.

A Figura 29 mostra as seguintes classes: classe Marcador, responsável pela concepção de novos marcadores, classe Cadastramento que permite a inserção de novos marcadores no sistema, através da execução do pacote Mk-Patt, que foi todo reestruturado e que corresponde ao módulo Cadastrador de Marcador, conforme visto na Figura 25. A classe Associação corresponde ao módulo Associador de Marcador.

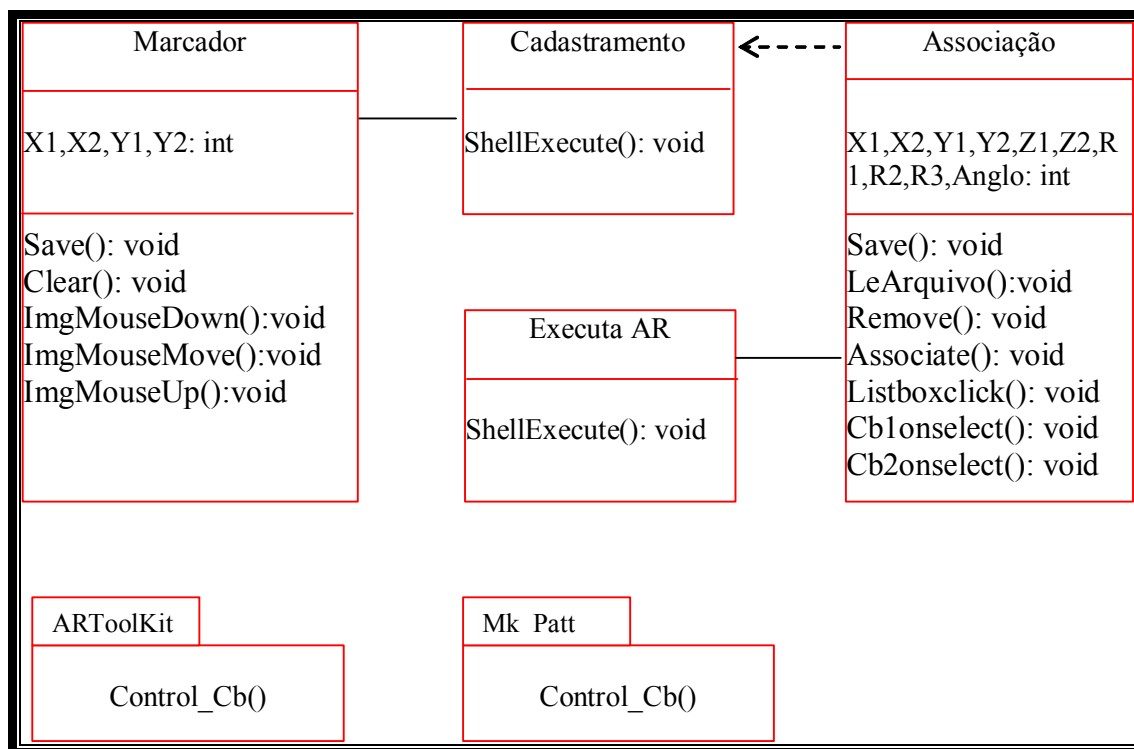


Figura 29. Diagrama de classes e pacotes.

A classe **Marcador** é utilizada para criação de marcadores através dos seguintes métodos: *ImgMouseDown()*, *ImgMouseMove()*, *ImgMouseUp()*, *Clear()*, *Save* e *OnChange()*.

O método *ImgMouseDown()*, é responsável pela captura dos valores dos atributos `x1` e `y1`(coordenadas linha e coluna), que armazenam os valores iniciais da imagem do marcador a ser desenhada. O método *ImgMouseMove()*, através do seu método particular *LineTo(x,y)*, responde pela trajetória do mouse do usuário. Com as coordenadas iniciais capturadas e a trajetória de desenho definida pelos métodos *ImgMouseDown()* e *ImgMouseMove()*, o método *ImgMouseUp()*, finaliza a construção da imagem do marcador, através das funções, *fillrect* e *ellipse*.

A classe **Cadastramento** é utilizada para executar o pacote Mk-Patt, que é responsável pelo registro de marcadores criados pela classe Marcador, esse pacote Mk-Patt é parte do software ARToolKit, utilizado para cadastramento de novos marcadores e que foi todo reestruturado conforme visto na Figura 25. A classe Associação é utilizada para associar um objeto em VRML com um marcador previamente cadastrado, essa classe é dependente da classe **Cadastramento**. Essa classe conta com os seguintes métodos: *LeArquivo()*, *Listbox1Click()*, *Remove()*, *Save()*, *Associate()*, *Cb1onselect()*, *Cb2onselect()*.

O método *LeArquivo()*, faz uma leitura do arquivo que contém todos os nomes dos marcadores associados com objetos VRML, o método *Listbox1Click()*, serve para capturar o

nome do arquivo. Esse arquivo contém atributos, tais como, translação, escala e rotação que podem ser modificados através da entrada de novos valores, o método *Save()* responde pelas alterações desses valores. O método *Remove()* faz a remoção de objetos virtuais e marcadores, enquanto que os métodos *Cb1onselect()*, *Cb2onselect()*, fazem a seleção dos objetos virtuais e marcadores a serem associados com o método *Associate()*. A classe Executa AR é responsável pela execução do pacote ARToolKit.

4.12.Considerações Finais

O *Borland C++ Builder* e a biblioteca GLUI permitiram o desenvolvimento das interfaces do sistema. Também foi visto a especificação do sistema através de diagramas de classes e casos de uso, no qual permitiu uma visualização geral do sistema, facilitando seu processo de desenvolvimento. No próximo Capítulo, serão abordados detalhes de implementação do sistema especificado.

Capítulo 5

5. DETALHES DE IMPLEMENTAÇÃO

5.1. Introdução

Neste Capítulo, são apresentados os detalhes de implementação de cada módulo do sistema visto no Capítulo anterior. Todas as interfaces dos módulos criados foram mostradas no Capítulo 4, por isso, aqui serão vistos apenas os detalhes de implementação do sistema.

5.2. Implementação do módulo Criador de Marcador

Esse módulo foi desenvolvido para permitir que o usuário crie um padrão ou marcador no qual permitirá a exibição de objetos virtuais.

Para a criação desse marcador, o usuário escolhe uma ferramenta de desenho, além de clicar e arrastar o mouse para a concepção desse novo marcador.

As ferramentas de desenho são componentes do *Borland C++ Builder* que visam facilitar as escolhas das ferramentas. Além disso, a interface desse módulo possui um componente *TImage*, onde as imagens são criadas. Após selecionar a ferramenta, o usuário pressiona o mouse sobre o componente *TImage* para a elaboração de uma Figura. Conforme já foi visto no Capítulo anterior, os métodos *ImgMouseDown()*, *ImgMouseMove()* e *ImgMouseUp()* são responsáveis por capturar as coordenadas *x1*, *y1*(linha, coluna), *x2* e *y2*(linha, coluna), além de utilizar os métodos *fillRect* e *ellipse* para a geração de um novo marcador. As Figura 30 e 31, mostram fragmentos de códigos dos métodos *imgMouseDown* e *imgMouseUp* respectivamente.

```
void __fastcall Tfrmcriarmarcador::imgMouseDown(TObject *Sender,
TMouseButton Button, TShiftState Shift, int X, int Y)

int a = x
int b = y
if(Pincel->Checked == true)
{
    img->Canvas->Pen->Color = frmcriarmarcador->Shpcor->Brush->Color;
    img->Canvas->Pen->Width = 5;
    img->Canvas->MoveTo(X,Y);
}
```

Figura 30. Fragmento de código do método *imgMouseDown*.

```

void __fastcall Tfrmcriarmarcador::imgMouseUp(TObject *Sender,
    TMouseButton Button, TShiftState Shift, int X, int Y)
{
    int c = x;
    int d = y;

    if(Ret->Checked == true)
    {
        TRect NewRect = Rect(a,b,c,d);
        img->Canvas->Brush->Color = clBlack;
        img->Canvas->FillRect(NewRect);
    }
    else if(Elipse->Checked == true)
    {
        //TRect NewElipse = Rect(a,b,c,d);
        img->Canvas->Brush->Color = clBlack;
        img->Canvas->Ellipse(a,b,c,d); //->FillRect(NewRect);
    }
}

```

Figura 31. Fragmento de código do método `imgMouseUp`.

Considerando os fragmentos de códigos das Figura 30 e 31 percebe-se que há quatro variáveis inteiras que armazenam as coordenadas do desenho, sendo $a = x1$, $b = y1$, $c = x2$ e $d = y2$. Através dos métodos *fillRect* e *ellipse*, juntamente com o método Canvas, é possível a geração de uma imagem Retangular ou Elíptica. O método Canvas, pode ser usado para desenhar linhas, retângulos, polígonos e Figuras, além disso, o Canvas usa o método *Brush* para preencher o interior da figura.

5.3. Implementação do módulo Cadastrador de Marcador

O ARToolKit possui um módulo chamado Mk-Patt, que é responsável pelo cadastramento de marcadores. No entanto, isso é feito de forma textual, ou seja, não tem uma interface que facilite esse processo de inclusão de novos marcadores no sistema, conforme foi visto na Figura 10. Por isso, desenvolveu-se uma interface que facilitasse esse processo, para isso, foi necessário reconstruir o módulo Mk-Patt do ARToolKit, conforme Figura 25.

O Mk-Patt utiliza uma biblioteca que contem uma função chamada *argInit2*, essa função utiliza um método chamado *glutCreateWindow* da biblioteca GLUT, que permite a criação de janelas. Esse método foi removido da função *argInit2* e inserido dentro do código principal do Mk-Patt. Isso foi necessário para ser possível construir uma sub-janela dentro da janela principal, conforme pode ser visto na Figura 33. Funções da biblioteca GLUI foram

inseridas para permitir a criação da interface para o Mk-Patt. O código da Figura 32 apresenta os seguintes métodos da biblioteca GLUT: *glutInitDisplayMode()*, *glutInitWindowPosition()*, *glutInitWindowSize()* e *glutCreateWindow()*. Esses métodos são responsáveis pelas configurações da janela, tais como, modo de abertura, posição, tamanho e criação da janela. Além da janela principal, foi necessário criar uma sub-janela, para permitir que o usuário entre com o nome do marcador de forma amigável.

```
glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGBA |  
GLUT_DEPTH | GLUT_STENCIL);  
glutInitWindowPosition(0, 0);  
glutInitWindowSize(CamAltura, CamLargura);  
  
// Cria Janela Principal  
JanelaPrinc = glutCreateWindow("Create Marker");
```

Figura 32. Fragmento de código inserido no Mk-Patt

Essa sub-janela foi criada a partir de métodos da biblioteca GLUI, a Figura 33 mostra esse fragmento de código.

```
glui = GLUI_Master.create_glui_subwindow (JanelaPrinc,  
GLUI_SUBWINDOW_TOP);  
edittext = glui->add_edittext("Enter Marker Name: ",GLUI_EDITTEXT_TEXT,text);  
edittext -> set_w(220);  
glui->add_column(true);  
glui->add_button("Save",1,control_cb);  
glui->add_column(true);
```

Figura 33. Fragmento de código inserido no Mk-Patt

O método *GLUI_Master.create_glui_subwindow (JanelaPrinc, GLUI_SUBWINDOW_TOP)*, permite a criação de uma sub-janela. Pode-se perceber que o primeiro argumento do método é o argumento da janela principal, chamado *JanelaPrinc*, e o segundo significa que a sub-janela será criada no topo. Conforme pode ser visto na Figura 33, foram utilizados métodos de complemento da interface, como *edittext*, *add_column* e *button*.

Ainda nessa Figura, o método *glui->add_button("Save",1,control_cb)*, além de colocar o botão *Save* na interface, executa a função *control_cb* toda vez que o botão *Save* for pressionado. A função *control_cb* foi implementada para que o nome e o conteúdo do

marcador fossem salvos e armazenados corretamente. A Figura 34 mostra o código da função *control_cb*.

```
void control_cb( int control )
{
    strcpy(string,edittext->get_text());
    if (arSavePatt(image, target, string) < 0)
        printf("ERROR!!\n");
    else
        glui->add_button("OK-Exit",0,(GLUI_Update_CB)exit);
}
```

Figura 34. Fragmento de código da função *control_cb*

Referente à função *control_cb*, quando o usuário posiciona o marcador em frente à câmera, entra com o nome e pressiona o botão Save, a condição if desta função verifica se o usuário digitou um nome pelo menos. Se o usuário digitou o nome do marcador, esse nome é armazenado e o botão OK-Exit é liberado para o usuário sair da interface. A interface desse módulo pode ser vista na Figura 25 do Capítulo 4, especificação do sistema. O próximo tópico detalha o módulo três do sistema.

5.4. Implementação do módulo Associador de Marcador

O item 2.6 (Capítulo 2), onde são apresentadas etapas de utilização do ARToolKit, foi visto como é o processo que o usuário deve fazer para utilizar esse software.

Esse é o último módulo desenvolvido e que permite ao usuário consolidar suas aplicações em RA.

A interface é composta por dois componentes do tipo *ComboBox*, que listam nomes de marcadores e nomes de objetos VRML respectivamente. Os nomes dos marcadores são advindos do processo de cadastramento visto no item 5.3, enquanto que os nomes dos objetos VRML são listados no *ComboBox* a partir de um diretório exclusivo chamado Data, onde os objetos e suas texturas devem estar previamente inseridos. Além disso, há mais dois componentes posicionados debaixo dos *ComboBoxs*, esses componentes servem para exibir a imagem do marcador e do objeto virtual, conforme ilustra a Figura 35.

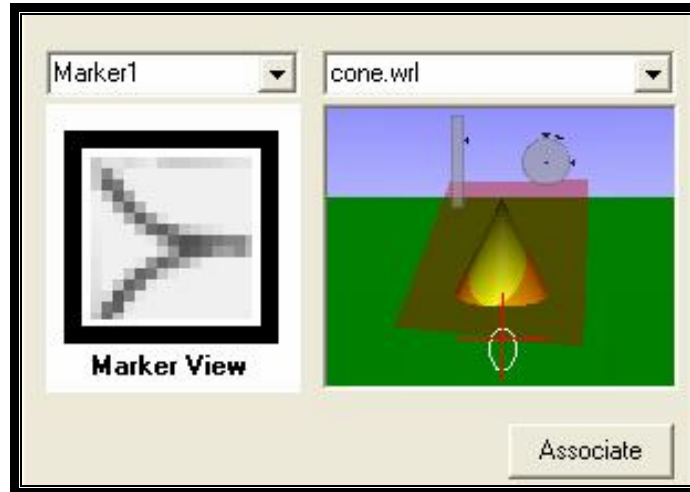


Figura 35. Imagem do marcador e objeto virtual

O componente *PaintBox(Marker View)*, é responsável pela exibição da imagem do marcador quando este for selecionado, a Figura 35 mostra à esquerda a imagem do marcador Marker1. Para poder exibir a imagem do marcador concomitantemente à sua seleção, foi necessário ler o conteúdo do arquivo marcador. Cada marcador é composto por um conjunto de números que estão em formato de matrizes 16 x 16. O trecho de código da Figura 36 mostra como é feito o processo de leitura do marcador.

```
int matriz[16][16], scala=5;
int i = cb1->ItemIndex; /* Pega índice de arquivo */
AnsiString caminho = pathdatageral + cb1->Items->Strings[i]; /* pega nome
arquivo */

arquivo = fopen(caminho.c_str(),"r");
for(int x=0;x<16;x++) {
    for(int y=0;y<16;y++){
        fscanf(arquivo,"%d",&matriz[x][y]); }}

for(int x = 0; x<16*scala; x++){
    for(int y = 0; y <16*scala; y++){
        PaintBox1->Canvas->Pixels[x][y] =
        RGB(matriz[x/scala][y/scala],matriz[x/scala][y/scala],matriz[x/scala][y/sc
        ala]); }}
```

Figura 36. Trecho de código do método LeArquivo

O método *LeArquivo*, faz a leitura do arquivo (marcador) que está sendo selecionado armazenando-o em uma matriz com dimensões 16 x 16. Após a leitura desse arquivo, é feita a divisão de cada pixel da matriz, por uma escala de 5. Isso foi feito para que a imagem pudesse ser visualizada, conforme foi mostrado na Figura 36 vista anteriormente. A exibição do objeto virtual à direita da imagem do marcador é feita a partir de um componente chamado

CppWebBrowser. Esse componente serve para mostrar o conteúdo de um *browser* (navegador. Ex: Internet Explorer). Os objetos em VRML são exibidos através de um navegador (*browser*) e um *plugin* previamente instalados. Enquanto o usuário seleciona um objeto virtual, no *ComboBox2*, é feito uma busca pelo nome desse objeto selecionado exibindo-o no componente *CppWebBrowser*.

O botão *Associate*, serve para associar o marcador e o objeto virtual selecionado. Porém, antes de fazer essa associação é feito uma busca para verificar se esse nome de marcador já não foi associado com um determinado objeto virtual. O sistema possibilita associar apenas um objeto virtual por marcador, essa condição é verificada por meio do trecho de código mostrado na Figura 37.

```
char string[30]="\0";
char aux3[30]="\0";
int p=i=flag=0;
if(cb1->Text != "" && cb2->Text != "")/*verifica se os campos estão vazios*/
{
    while(i < k)
    {
        if(cb1->Text == Str1->Strings[i])/*Compara os padrões já cadastrados */
        {
            flag = 1;
            i = k;
        }

        i++;
    }
}
```

Figura 37. Trecho de código do botão Associate.

O comando condição do código verifica se o usuário selecionou algum objeto, caso não tenha selecionado é exibido uma mensagem alertando que algum objeto deve ser selecionado.

O laço é responsável pela comparação do marcador selecionado pelo usuário com cada marcador já associado com algum objeto virtual. O componente *ListBox* exibe todas as ligações já existentes entre marcadores e objetos virtuais, conforme pode ser visto na Figura 38.

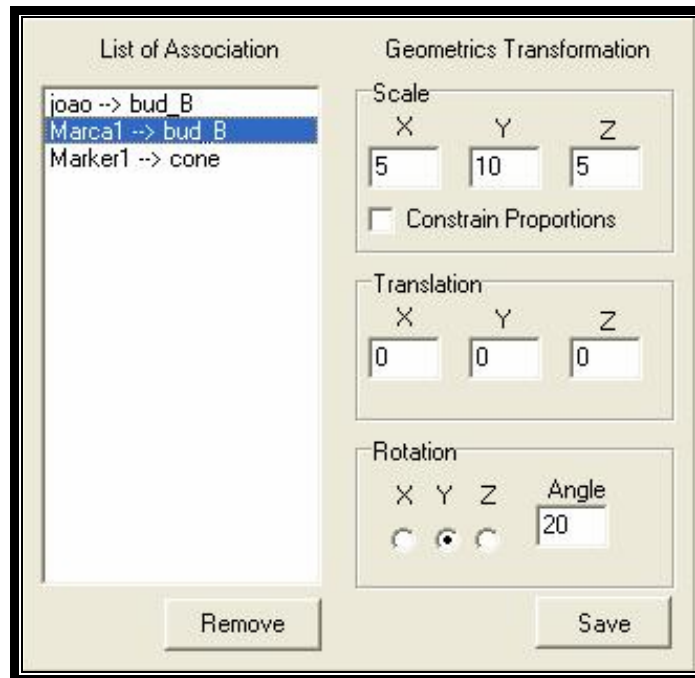


Figura 38. Parte da tela da interface que associa marcadores.

Do lado esquerdo da Figura é mostrado em *List of Association*, os nomes dos marcadores e dos objetos virtuais, que estão associados entre si. Esta lista de associações foi implementada no método *Associate()*, através de uma Lista duplamente encadeada, onde é possível inserir e remover objetos em qualquer ponto dessa lista.

Conforme pode ser observado na Figura 38, foi implementado também as transformações geométricas escala, translação e rotação dos objetos. Quando um usuário seleciona um objeto na lista de associações, todas as transformações do objeto selecionado são exibidas, permitindo alterações se necessário. O método *Save()* é responsável por registrar qualquer alteração feita nos objetos.

O método *Remove()*, serve para remover algumas associações indesejadas pelo usuário, esse método verifica na Lista implementada no método *Associate()*, os nomes dos arquivos que serão removidos.

Para programar o mecanismo de exibição da imagem do marcador quando selecionado, foi necessário fazer um estudo para saber como é feito o armazenamento desse arquivo marcador.

5.5. Considerações Finais

O sistema desenvolvido possibilita ao usuário uma melhor utilização do software ARToolkit por usuários que desconhecem os passos de execução do mesmo, e pode por esta razão, ser uma forma de promover a utilização do ARToolkit. A integração da biblioteca GLUI com a biblioteca ARToolkit permitiu a construção de um dos módulos da interface que possibilita a inserção de marcadores no sistema. As interfaces foram desenvolvidas no C++ builder o que possibilitou uma forma amigável para a construção das aplicações.

O próximo Capítulo apresenta os resultados, limitações e conclusões do sistema proposto, baseado em experimentos realizados com o mesmo.

Capítulo 6

6. RESULTADOS E LIMITAÇÕES

6.1. Introdução

Este Capítulo apresenta uma breve explicação sobre o funcionamento do software na visão de utilização pelo usuário. São abordadas questões sobre avaliação do sistema, através de testes básicos sobre usabilidade.

6.2. Utilização do Sistema

Ao executar o sistema o usuário se depara com uma interface composta por quatro botões, que dão acesso às etapas necessárias para iniciar o uso do sistema.

- *Create Marker*
- *Register Marker*
- *Associate Marker*
- *Execute Application AR*

Cada etapa mostrada acima possibilita ao usuário realizar uma determinada ação. A primeira etapa permite que o usuário crie novos marcadores através de uma interface amigável, enquanto que, o botão *Register Marker* possibilita a inserção desses novos marcadores no sistema.

O botão *Associate Marker* permite que o usuário faça associações entre marcadores e objetos virtuais, além disso, é possível ainda, alterar as transformações geométricas desses objetos. Por fim o botão *Execute Application AR* possibilita a execução de um dos módulos do ARToolKit que permite a visualização da aplicação sobre o marcador. A imagem de cada interface foi mostrada no Capítulo 4.

6.3. Avaliação do Sistema

Com base em preceitos da ISO/NORM 9241-10 Usability (conjunto de normas que permite avaliar a capacidade de um sistema interativo oferecer ao usuário a possibilidade de realizar tarefas de maneira eficaz e agradável (Prugmper, 1999)).

De acordo com essas normas foi elaborado um questionário de avaliação do sistema disponível no Anexo. Este questionário tem por objetivo verificar questões sobre usabilidade do sistema desenvolvido.

O sistema foi apresentado a diversos usuários potenciais, para avaliar as questões relacionadas com a interface e usabilidade. Primeiramente, foram fornecidas informações sobre o que é Realidade Aumentada, como se utiliza o sistema e, em seguida, de forma individual, eles iniciaram o uso da ferramenta. Após a execução, responderam o questionário de avaliação. Os alunos do quinto ano de Engenharia de Computação foram os usuários que participaram da avaliação do sistema, esses usuários nunca haviam utilizado o ARToolkit ou qualquer outro software de RA.

6.4. Satisfação quanto ao uso do sistema

- Facilidade de uso do sistema

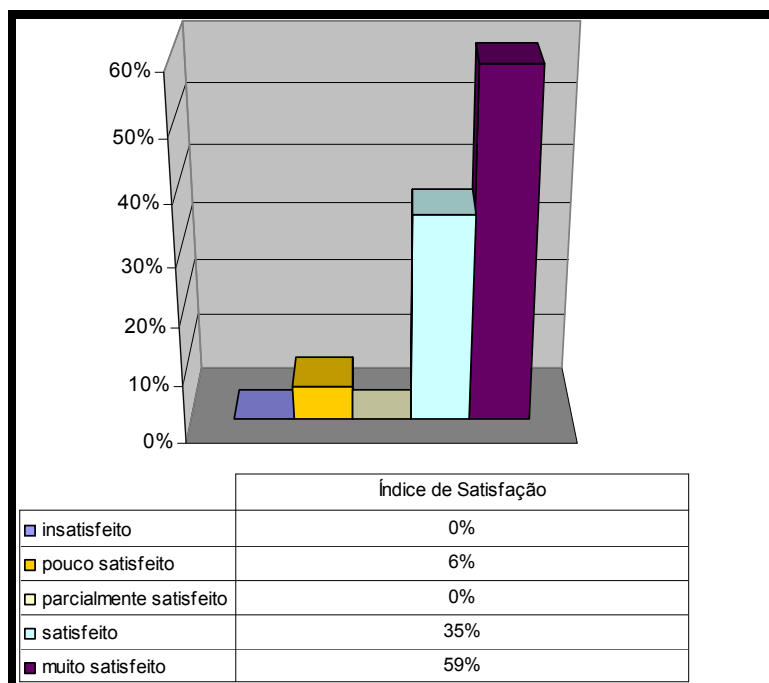


Figura 39. Facilidade de uso do sistema

- Controle sobre a atividade (possibilidade de interromper, suspender, etc)

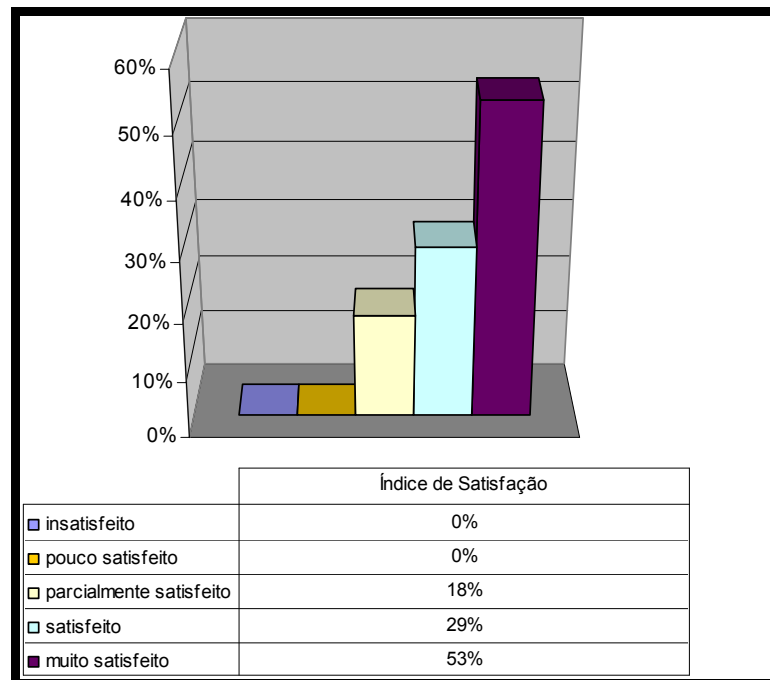


Figura 40. Controle sobre a atividade

- Possibilidade de visualização clara do que está sendo executado

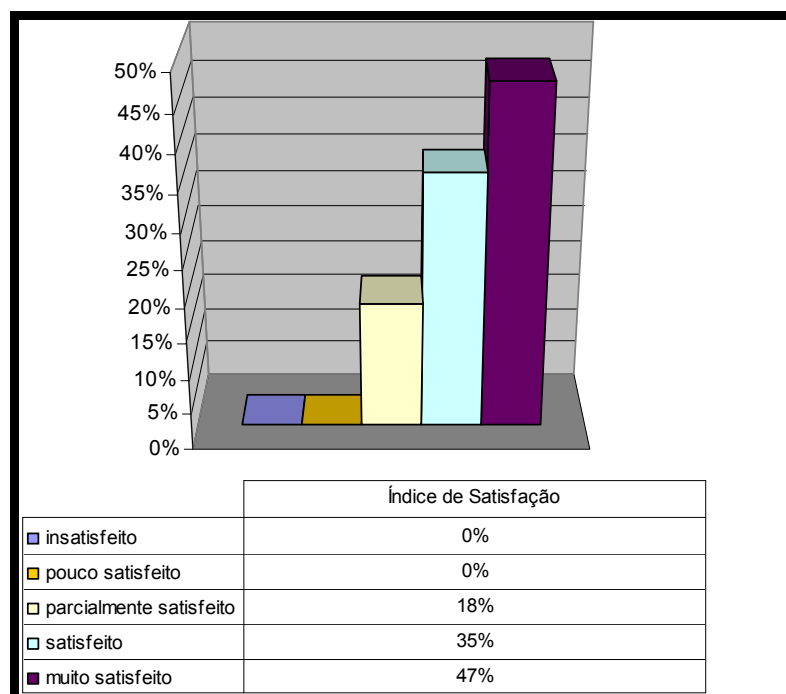


Figura 41. Possibilidade de visibilização clara do que está sendo executado

- Redução de trabalhos repetitivos

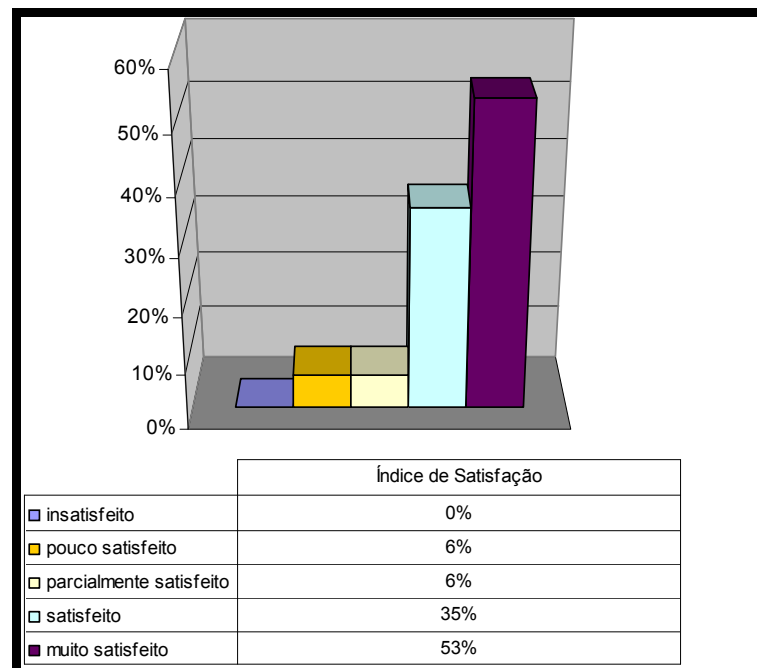


Figura 42. Redução de trabalhos repetitivos

6.5. Quanto à manipulação de marcadores e objetos virtuais

- Facilidade de concepção de um novo marcador

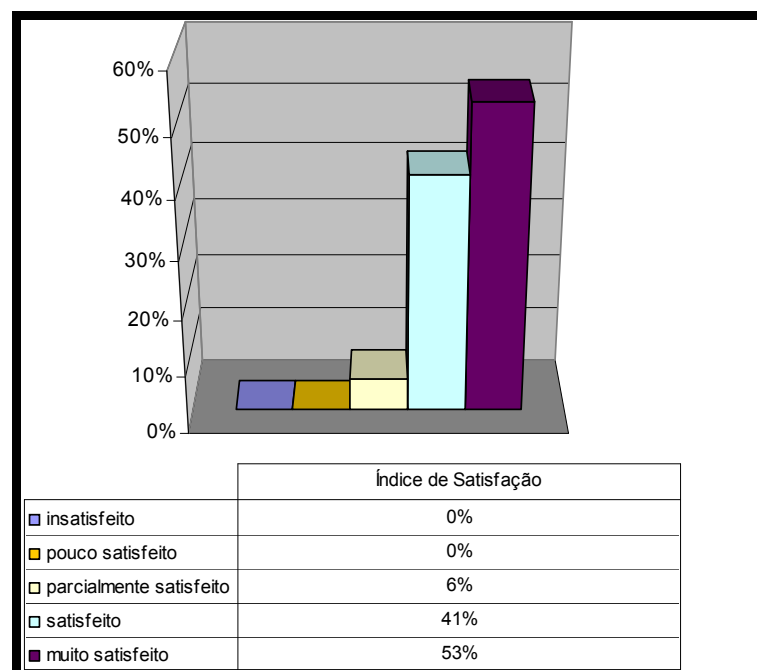


Figura 43. Facilidade de Concepção de um novo marcador

- Facilidade de inserção de um novo marcador no sistema

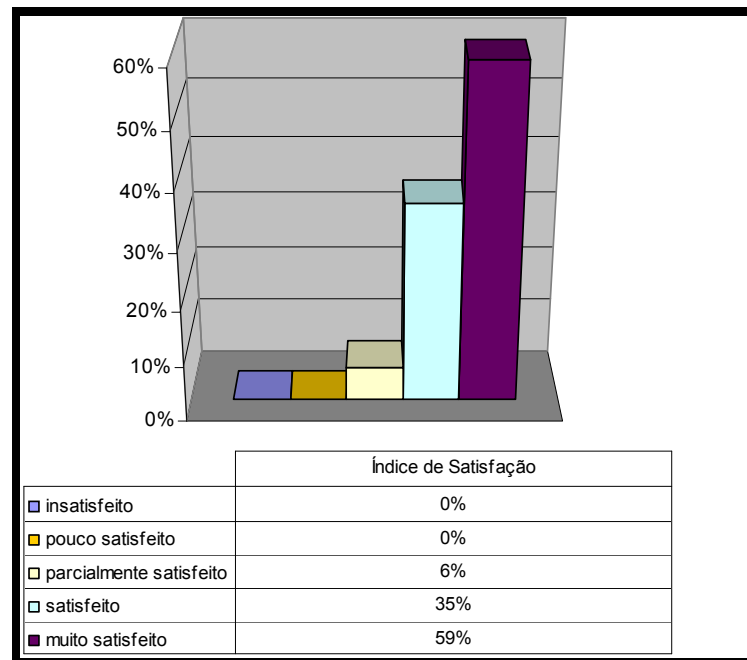


Figura 44. Facilidade de inserção de marcador

- Facilidade de associar um objeto virtual em um marcador

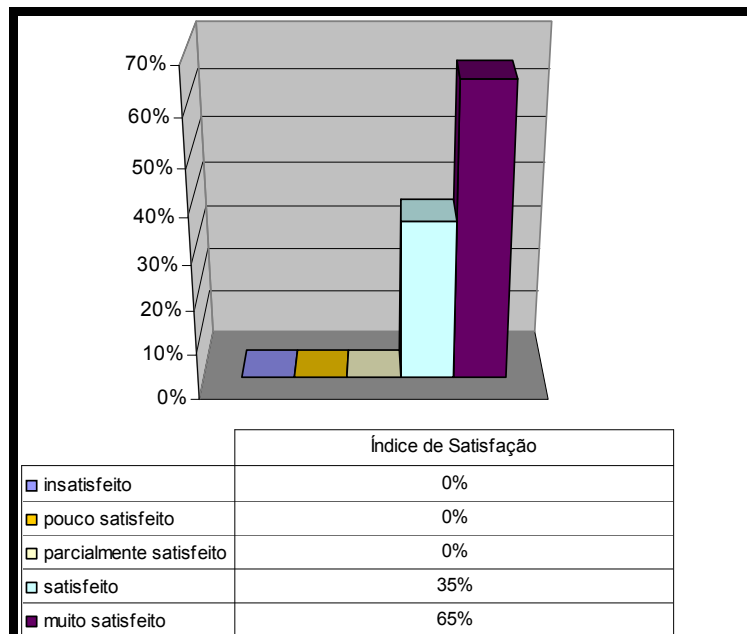


Figura 45. Facilidade para associar um objeto virtual em um marcador

- Facilidade para remover um objeto virtual de um marcador

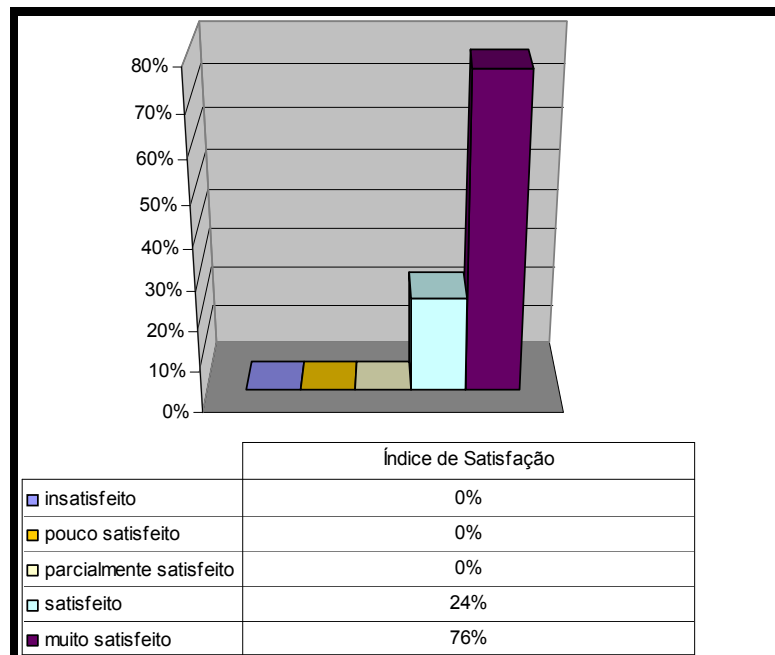


Figura 46. Facilidade para remover um objeto de um marcador

6.6. Limitações do Sistema

Os tópicos a seguir mostram algumas limitações do sistema.

- Objetos virtuais não podem ser associados aos marcadores em tempo de execução.
- Os objetos virtuais são inseridos manualmente pelos usuários.
- O sistema não trabalha com objetos virtuais desenvolvidos em OpenGL.

6.7. Considerações Finais

Este Capítulo apresentou de forma resumida o funcionamento do sistema, a avaliação e suas limitações. Apesar de algumas limitações o sistema se mostrou adequado e atendeu os objetivos propostos.

Capítulo 7

7. CONCLUSÕES

O desenvolvimento de interfaces de usuário efetivas, úteis, seguras e agradáveis são fatores importantes para o sucesso de um determinado software. Cada vez mais as pessoas necessitam de interfaces que absorvam seus conhecimentos, habilidades e propósitos.

Esta dissertação descreveu estudos que englobam projetos de interfaces de usuários e sistemas de RA que utilizam a biblioteca ARToolKit. Este estudo teve como objetivo o desenvolvimento de um sistema com interfaces que pudesse oferecer suporte aos usuários na construção de aplicações de RA.

Durante a fase de pesquisa verificou-se a existência de vários sistemas de RA, apoiados no uso do ARToolkit, sendo que a maioria deles apresenta aplicações específicas. Não foi encontrado nenhum sistema que tivesse uma interface para facilitar o processo de criação de marcadores, inserção, associação de objetos virtuais a esses marcadores, ou seja, o uso do software ARToolKit por usuários comuns. Além disso, foi descrito de forma detalhada o estado da arte dessa biblioteca, mostrando todas as etapas que um usuário necessita saber para utilizá-lo.

De acordo com a avaliação dos usuários, pôde se perceber que o sistema apresentou um bom índice de satisfação quanto às questões relacionadas com sua usabilidade. Muitos usuários ao utilizar o ARToolKit se perdiam na construção de uma aplicação. Já na utilização da interface os usuários conseguiam obter uma determinada aplicação sem muitas dificuldades se comparado com a forma tradicional de utilização do ARToolKit.

Algumas dificuldades podem ser destacadas durante o desenvolvimento desse sistema. O módulo Cadastrador de Marcador foi feito através da reconstrução do aplicativo Mk_Patt. Para isso, foi necessário identificar o funcionamento desse aplicativo, para poder modificá-lo adicionando uma sub-janela, a fim de permitir que usuários pudessem entrar com o nome do marcador de forma intuitiva. A descoberta da implementação do Mk_Patt foi trabalhosa devido a falta de documentação. Como o Mk-Patt foi desenvolvido usando o ambiente *Microsoft Visual C++*, essa sub-janela foi construída a partir desse ambiente com uso da biblioteca GLUI. A criação dos outros módulos do sistema foi feito utilizando o *Borland C++ Builder*, devido à facilidade de componentes gráficos encontrados nesse ambiente.

Uma outra dificuldade encontrada refere-se à da definição da interface, a princípio o módulo Criador de Marcador e Executor RA estava dentro de uma única janela, através de dois botões o usuário trocava de um módulo para outro. Com esses dois botões na interface, o usuário teria dificuldades na visualização de uma aplicação em RA, caso fosse utilizado um HMD. Por isso, optou-se por desenvolver uma interface principal, onde o usuário pudesse acessar todos os outros módulos de forma separada, inclusive os módulos Criador de Marcador e Executor RA.

Através desse sistema de janelas desenvolvido para a biblioteca ARToolKit, fundamentado nos conceitos de interfaces, foi permitido aos usuários executarem os mecanismos básicos para desenvolver uma aplicação de RA.

7.1. Trabalhos Futuros

Como trabalhos futuros, podem ser sugeridas as seguintes implementações:

1. O sistema deve permitir que objetos virtuais sejam associados aos marcadores em tempo de execução, isso pode ser feito através do uso de oclusão, além disso alguns parâmetros podem ser modificados através de marcadores auxiliares de controle.

A iluminação é um problema para essa implementação, pois qualquer variação na luz pode ser confundida com oclusão fazendo com que objetos virtuais sejam adicionados sem intenção do usuário.

2. Desenvolvimento de um mecanismo de inserção de objetos virtuais dentro do sistema de forma visual.

Esse mecanismo não foi implementado por questões de tempo. No entanto, esse mecanismo pode ser feito através da importação de todos os objetos virtuais para um diretório específico. O problema é que, além dos objetos virtuais é necessário importar as texturas para o mesmo diretório.

3. Desenvolver dentro do módulo criar marcador um mecanismo para validação, este mecanismo tem como intuito evitar marcadores simétricos e desenhos que ultrapassam as bordas laterais desses marcadores.

REFERÊNCIAS BIBLIOGRÁFICAS

ARtag. Disponível em: <<http://www.artag.net/>> . Acesso em: 06 jul. 2007.

ARToolKit. Disponível em: <<http://www.hitl.washington.edu/artoolkit/>>. Acesso em: 08 mar. 2005.

Azuma, R. T., A survey of augmented reality, Presence: Teleoperators and virtual environments, vol. 6, p. 355- 385. 1997.

Bevan. Estudo de Usabilidade em Sistemas Móveis com Foco em PDAs. 2005.144f. Dissertação (Mestrado Profissional) - Instituto de Computação Universidade Estadual de Campinas, Campinas, 2005.

Billinghurst, M. Augmented Reality in Education. Disponível em: <<http://www.newhorizons.org/strategies/technology/billinghurst.htm>>. Acesso em: 22 maio. 2005.

Constantine, L. What do users want? Engineering usability into software. Windows Tech Journal, 4 (12), pp. 30-39, December 1995.

Ferreira, A. B. Dicionário Aurélio Eletrônico Século XXI. Rio de Janeiro: Nova Fronteira. 1999.

GLOBO. Globo.com. Disponível em: <<http://www.globo.com>> . Acesso em: 07 jun. 2007.

GLUT. GLUT - The OpenGL Utility Toolkit. Disponível em: <<http://www.opengl.org/resources/libraries/glut/>> . Acesso em: 30 jun. 2007.

Kato, H., M. Billinghurst, et al. ARToolKit version 2.33: A software Library for Augmented Reality Applications. Disponível em: <<http://www.hitl.washington.edu/Artoolkit/>>. Acesso em: 10 maio 2005.

Kirner, C(1999). Sistemas de Realidade Virtual. Disponível em: <<http://www.dc.ufscar.br/~grv/tutrv/tutrv.htm>>. Acesso em: 14 jan. 2006.

Kirner, C.; Siscoutto, R, A. Fundamentos de Realidade Virtual e Aumentada. In: Kirner, C.; Siscoutto, R, A. Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações. Petrópolis: SBC – Sociedade Brasileira de Computação, cap. 1, p. 2-21. 2007.

Lima, P. Interfaces homem máquina. Disponível em: <<http://www2.uepa.br/paulolima/disciplinas/ufpa.php>>. Acesso em: 5 fev. 2007.

Maddix, F. Human-computer interaction: theory and practice. England: Ellis Horwood Limited, 1990.

Microsoft, 2007. Introdução ao DirectX. Disponível em: <<http://www.microsoft.com/brasil/msdn/Tecnologias/arquitetura/DirectX.msp>> Acesso em: 29 jun. 2007.

Microsystems, 2001. Microsystems, S. Object-Oriented Application Analysis and Design for Java TM Technology (UML) OO-226 Student Guide. [S.l.]: Sun Microsystems, 2001.

Milgram, P. e J. H. Colquhoun. A Taxonomy of Real and Virtual World Display Integration. Disponível em: <http://vered.rose.utoronto.ca/publication/1999/Milgram_Colquhoun_ISMR1999.pdf>. Acesso em: 02 jul. 2006.

Mogilev, D. et al. AR Pad: an interface for face-to-face AR collaboration. CHI '02 extended abstracts on Human factors in computing systems, ACM Press, New York, USA, p. 654 – 655, Apr, 2002.

Norman, D. A, 2002. The Design of Everyday Things: Basic Books, 2002.

OpenGL, 2007. OpenGL Overview. Disponível em: <<http://www.opengl.org/about/overview/>>. Acesso em: 08 jun. 2007.

OSGART, 2007. OSGART ARToolKit for OpenSceneGraph. Disponível em: <<http://www.artoolworks.com/community/osgart/index.html>>. Acesso em: 18 jun. 2007.

Paradyme, 2007. User experience design. Disponível em: <<http://www.paradymesolutions.com/articles/what-is-user-experience-design/>>. Acesso em: 07 jun. 2007.

Poupyrev, I et al. Developing a Generic Augmented Reality Interface. IEEE Computer Society, vol. 35, no. 3, p. 44-50, Mar, 2002.

Pressman, R. S. Engenharia de Software. São Paulo: McGraw-Hill, 720 p. 2006.

Prugmper. Test it: ISONORM 9241/10. In: H.-J. BULLINGER & J. ZIEGLER Eds. Proceedings of HCI International, Munich, 22-27 August 1999. Mahwah, NJ: Lawrence Erlbaum. 1999.

Rademacher, P. GLUI User Interface Library Versão 2.1. Disponível em: <<http://www.cs.unc.edu/~rademach/glui/>>. Acesso em: 24 mar 2005

Radfahrer, apud Fernandes 2004. Usabilidade de Interfaces Universitárias na Web. 2004. 149f. Monografia (Ciência da Computação) – Centro das Ciências Agrárias e das Engenharias. Universidade do Sul de Santa Catarina – UNISUL – Palhoça, 2004.

Rekimoto, J., Nagao, K. The World through the Computer: Computer Augmented Interaction with Real World Environments, Proc of UIST '95, (ed. B.A.Myers), ACM Press, Pennsylvania, p 29-36, 1995.

Scapin, 1993. Organizing Human Factors Knowledge for Evaluation and Design of Interfaces - B.P. 105/78153 Le Chesnay Cedex, France.

Shneidermann, B. Designing the user interface: strategies for effective human-computer interaction. Los Angeles: Addison Wesley, 1987.

Spitzak, M. Programming Manual FLTK. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA, Junho 1991.

Sun Microsystems Java 3D Engineering Team, 2007. Java 3D API Tutorial. Disponível em <<http://developer.java.sun.com/developer/onlineTraining/java3d/>>. Acesso em: 08 jun. 2007.

Tsuchimoto, T *et al.* Augmented Reality Interface for Measurement of Tissue Stiffness. Proceedings of the 7th Int. Conf. on Human and Computer HC 2004, Aizu-Wakamatsu, p. 148-153, Sep, 2004.

Valente, E. C, 2004. Padrões de Interação e Usabilidade. 2004. 73f. Dissertação (Mestrado Profissional) - Instituto de Computação Universidade Estadual de Campinas, Campinas, 2004.

Wisner, A. Ergonomia: método e técnica. São Paulo: FTD/OBORÉ. 1987.

ANEXO

Avaliação do Sistema

Avaliador: _____ Data: ----/----/2007

Grau de instrução:

a) Satisfação quanto ao uso do sistema

- 1) Quanto à clareza dos termos utilizados (em: botões, menus, mensagens do sistema, campos de preenchimento).

Pouco ☐ ☐ ☐ ☐ ☐ Muito

- 2) Número de funções existentes no sistema

Pouco ☐ ☐ ☐ ☐ ☐ Muito

- 3) Facilidade de uso do sistema

Pouco ☐ ☐ ☐ ☐ ☐ Muito

- 4) Confiança de que o sistema já executou o que foi pedido

Pouco ☐ ☐ ☐ ☐ ☐ Muito

- 5) Número de passos necessário para a realização de uma atividade

Pouco ☐ ☐ ☐ ☐ ☐ Muito

- 6) Controle sobre a atividade (possibilidade de interromper, suspender, etc)

Pouco ☐ ☐ ☐ ☐ ☐ Muito

- 7) Possibilidade de visualização clara do que está sendo executado

Pouco ☐ ☐ ☐ ☐ ☐ Muito

b) Satisfação quanto à melhoria na rotina de trabalho

8) Facilidade de alterar dados já fornecidos

Pouco ☐ ☐ ☐ ☐ ☐ Muito

9) Redução de trabalhos repetitivos

Pouco ☐ ☐ ☐ ☐ ☐ Muito

10) Resgate de informações já fornecidas ao sistema

Pouco ☐ ☐ ☐ ☐ ☐ Muito

c) Quanto à manipulação de marcadores e objetos virtuais

11) Facilidade de concepção de um novo marcador

Pouco ☐ ☐ ☐ ☐ ☐ Muito

12) Facilidade de inserção de um novo marcador no sistema

Pouco ☐ ☐ ☐ ☐ ☐ Muito

13) Facilidade de associar um objeto virtual em um marcador

Pouco ☐ ☐ ☐ ☐ ☐ Muito

14) Facilidade para remover um objeto virtual de um marcador

Pouco ☐ ☐ ☐ ☐ ☐ Muito