



UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS – UNIPAC
FACULDADE DE TECNOLOGIA E CIÊNCIAS DE CONSELHEIRO
LAFAIETE
CURSO DE ENGENHARIA DA COMPUTAÇÃO

LEONARDO PEREIRA SANTOS

DEMONSTRADOR VIRTUAL DE AMBIENTES VIA WEB COM
RECURSOS DE REALIDADE AUMENTADA

CONSELHEIRO LAFAIETE

2011



UNIVERSIDADE PRESIDENTE ANTÔNIO CARLOS – UNIPAC
FACULDADE DE TECNOLOGIA E CIÊNCIAS DE CONSELHEIRO
LAFAIETE
CURSO DE ENGENHARIA DA COMPUTAÇÃO

LEONARDO PEREIRA SANTOS

DEMONSTRADOR VIRTUAL DE AMBIENTES VIA WEB COM
RECURSOS DE REALIDADE AUMENTADA

Monografia apresentada ao curso de Engenharia da Computação da Universidade Presidente Antônio Carlos – UNIPAC, como requisito parcial para obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Jean Carlo Mendes

CONSELHEIRO LAFAIETE

2011

Leonardo Pereira Santos

**DEMONSTRADOR VIRTUAL DE AMBIENTES VIA WEB COM RECURSOS DE
REALIDADE AUMENTADA**

Monografia apresentada à Universidade Presidente Antônio Carlos – UNIPAC, como requisito parcial
para obtenção do título de Bacharel em Engenharia da Computação.

BANCA EXAMINADORA

Prof. Jean Carlo Mendes (Orientador)

Universidade Presidente Antônio Carlos – UNIPAC

Prof. Kléber Netto Fonseca

Universidade Presidente Antônio Carlos – UNIPAC

Aprovada em: ___/___/___

Dedico esse trabalho à minha mãe Judith, e à minha irmã Jéssica, meus exemplos, e minha fonte de apoio incondicional em todos os momentos.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela oportunidade a mim concedida de poder estar concluindo a graduação.

À minha mãe por ser a principal responsável pela minha escolha da área de computação e estar hoje consciente de que fiz a escolha mais feliz e certa para a minha carreira profissional e realização pessoal. Agradeço também a ela pelo grande apoio, estando sempre presente aconselhando-me e incentivando-me a realizar o meu ideal.

À minha querida irmã, Jéssica, pela sincronia de pensamentos e palavras positivas, sempre me motivando a avançar em busca da realização do meu ideal, fazendo-me acreditar que seria capaz de realizar os meus sonhos.

Ao meu pai, pelas orações e apoio.

De maneira especial, aos meus tios Hélio e Maria Amélia e minha querida prima Daniela pelo apoio, conselhos e orações ao meu favor e pelo presente do meu primeiro computador, que foi meu fiel companheiro de estudo nos períodos iniciais do curso.

Ao meu amigo Gil, pelo companheirismo durante o curso e pela nossa parceria na ideia e no desenvolvimento deste trabalho.

Ao Rodrigo, pela importante ajuda no desenvolvimento do código da minha aplicação e concretização do meu trabalho.

Aos demais familiares e amigos que foram marcantes com o apoio e orações ao meu favor, e que tenho certeza de que compartilham comigo mais essa minha vitória.

Ao meu orientador Jean Carlo pela orientação e incentivo a mim dedicados, sempre com uma pitada de humor (NaN), motivando-me para a concretização de mais este objetivo.

À equipe TI da CSN: Márcio, Rogério, Marcelino, Lucas, André, Cezanne, Marcus, Maykon, Leonardo, Luciano, Felipe, Fernando e Vítor, pelo exemplo técnico e profissional, pelo companheirismo e contribuição para o meu crescimento profissional.

Agradeço aos professores Rone Ilídio, Luciana Carla, Emerson Tavares, Daniella Barros, Kléber Fonseca, Alfredo Ganime e Wesley Luciano, pelo conhecimento transmitido, e exemplo profissional, que me foram de grande valia no desenvolvimento deste trabalho.

Agradeço aos demais colegas de classe, especialmente ao Wesley Polidoro, Shirley Basílio e Antônio José, pela parceria e os bons momentos compartilhados em classe.

“The product is no longer the basis of value. The experience is.”

Venkat Ramaswamy

*“Nós estamos vivendo na economia da experiência e o cliente é a estrela do show.
Se eu vou gastar milhares de dólares em alguma coisa, quero ter a experiência de isso
ser um conto de fadas.”*

Milton Pedraza

RESUMO

A monografia trata da implementação de um sistema de um Demonstrador virtual de ambientes via web com recursos de Realidade Aumentada. O sistema visa suprir a necessidade de exibição de algum ambiente em aplicações web, como por exemplo, a demonstração de um imóvel ou um projeto arquitetônico. Para isso, o usuário visualiza e interage com um ambiente virtual 3-D. Como forma de aumentar a interação com o usuário, o sistema possui recursos de reconhecimento angular e rotacional do marcador da Realidade Aumentada.

Palavras-chave: Realidade Aumentada, 3-D, Interatividade

ABSTRACT

The monograph consists in implementing a system of a virtual web Demonstrator of environments with Augmented Reality resources. The system aims to fill a need to display some web application environment, such as the demonstration of a building or an architectural design. For this, the user sees and interacts with a 3-D virtual environment. The system simulates a building in 3-D with which the user interacts through Augmented Reality. In order to increase the user interaction, the system has features of angular and rotational recognition of the Augmented Reality marker.

Keywords: Augmented Reality, 3-D, Interactivity

LISTA DE FIGURAS

Figura 1 – Sensorama	18
Figura 2- Diagrama de realidade/virtualidade contínua.	20
Figura 3 - Exemplos de uso da Realidade Aumentada na publicidade	21
Figura 4 - Diagrama de funcionamento do ARToolkit.....	27
Figura 5 – Imagem capturada pela câmera	28
Figura 6 - Imagem limiarizada	28
Figura 7 - Imagem binarizada.....	29
Figura 8 - Marcadores encontrados pelo FLARToolKit	29
Figura 9 - Área do marcador utilizada para comparação.....	30
Figura 10 - Tabela de comparação dos marcadores identificados.....	31
Figura 11 - Destaque para o marcador reconhecido pelo FLARToolkit.....	31
Figura 12 - Relacionamento entre as coordenadas do marcador e da câmera.....	32
Figura 13 - Objeto virtual integrado ao ambiente real	32
Figura 14 – Exemplo de código de barras	35
Figura 15 – Exemplo de um QR Code contendo um texto.....	36
Figura 16 – Direções das informações dos códigos	36
Figura 17 – Leitura de um QR Code por um Celular	37
Figura 18 - Exemplos de marcadores fiduciais	37
Figura 19 - Marker Generator em funcionamento.....	38
Figura 20 – Exemplos de marcadores corretos e incorretos	39
Figura 21 – Coordenadas do marcador.....	39
Figura 22 - Cubo montável com marcadores representando os diversos ambientes.....	41
Figura 23 - Cubo montado.....	42

Figura 24 – Tela de confirmação de acesso à câmera	42
Figura 25 - Diagrama do fluxo de funcionamento do sistema	43
Figura 26 - Estrutura das pastas do projeto no FlashDevelop	44
Figura 27 - Função FLARCode com parâmetros padrões	46
Figura 28 - Função onFrameCapturado	47
Figura 29 - Função onFrameCapturado	47
Figura 30 - Função MovimentoXYZ.....	48
Figura 31 - Função verificarAcoes	49
Figura 32 – Tela com os valores de posição e a ação a ser realizada	50
Figura 33 - QRCode para acesso à aplicação Web.....	50
Figura 34 - Aplicação Web.....	51
Figura 35 - Aplicação Web ativa exibindo o estádio Mineirão	51

LISTA DE TABELAS

Tabela 1 – Interfaces de interação computacionais	24
Tabela 2 – Comparação entre as bibliotecas de Realidade Aumentada	27
Tabela 3 - Classes e formatos suportados pelo Papervision3D	33
Tabela 4 – Capacidades de armazenamento de um QR Code	36
Tabela 5 – Ferramentas instaladas	44
Tabela 6 – Classes básicas do FLARToolkit	45
Tabela 7 – Padrão de atribuição de movimento	48

LISTA DE ABREVIATURAS E SIGLAS

3D – Três dimensões

API - *Application Programming Interface* (Interface de Programação de Aplicativos)

AS3 – *ActionScript 3*

CS5 – *Creative Suite 5*

iOS – iPhone OS (Sistema Operacional para iPhone)

PDF – *Portable Document Format*

QRCode – *Quick Response code* (Código de Resposta Rápida)

RA – Realidade Aumentada

RF – Requisito Funcional

RIA – *Rich Internet Application* (Aplicação Rica de Internet)

RNF – Requisito não-Funcional

RV – Realidade Virtual

RVA – Realidade Virtual e Aumentada

SO – Sistema Operacional

TV – Televisão

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Justificativa.....	15
1.2. Objetivos	16
1.2.1. Objetivo Geral	16
1.2.2. Objetivos Específicos	16
1.3. Estrutura da Monografia.....	16
1.4. Metodologia.....	17
2. REFERENCIAL TEÓRICO	18
2.1. Realidade Virtual.....	18
2.2. Ambiente Virtual.....	19
2.3. Realidade Misturada.....	19
2.4. Realidade Aumentada.....	20
2.5. Diferenças entre Realidade Virtual e Realidade Aumentada.....	22
2.6. RIA – Rich Internet Application.....	23
2.7. Interatividade.....	23
2.7.1. Interface.....	23
2.7.2. Interação Humano – Computador (IHC)	24
2.8. Adobe Flash.....	25
2.9. FlashDevelop.....	25
2.10. ActionScript3.....	25
2.11. ARToolkit.....	26
2.12. FLARToolkit.....	27
2.13. Ferramentas 3D	33
2.13.1. Papervision3D	33
2.13.2. Google SketchUp.....	34
2.13.3. Blender3D.....	34
2.14. Códigos Interativos.....	34
2.14.1. Códigos de barras	35
2.14.2. QR Code	35
2.15. Marcador	37
3. Desenvolvimento	40
3.1. Interface de interação.....	40
3.2. Ferramentas utilizadas	43
3.3. Classes	45
3.4. Padrões Utilizados	45
3.5. Funções criadas	46
3.6. Testes.....	49
3.7. Aplicação Web	50
3.8. Testes finais.....	51
4. CONSIDERAÇÕES FINAIS.....	52
4.1. Conclusão	52
4.2. Trabalhos futuros.....	52
5. REFERENCIAS	54
APÊNDICE A – Engenharia de Software do sistema	56

1.	Requisitos funcionais.....	56
2.	Requisitos Não funcionais	57
3.	Limitações do sistema	57
4.	Requisitos de desempenho:	57
5.	Diagrama de Casos de Uso.....	58

1. INTRODUÇÃO

Desde a antiguidade o homem busca formas de interação com as pessoas, objetos e ferramentas de trabalho. Até a idade média, tal interação era obtida através de métodos artesanais, tais como pintura e outros métodos como a escrita e a fala. Após a Revolução industrial, o homem foi obrigado a interagir com máquinas, abstraindo funções de botões e alavancas. Após o advento da computação, foi introduzido um novo modelo de interação, mais abstrato, onde se tornou necessário um treinamento mais completo, pois passava a deixar de lado o mundo real para tratar do mundo virtual.

Tal modelo, apesar dos benefícios, se tornou restrito a um grupo seletivo de pessoas, pois necessitava grande conhecimento e capacidade elevada de adaptação com o sistema.

Várias tentativas foram feitas para mudar este quadro. Os pesquisadores passaram a buscar meios para o sistema se adaptar ao usuário. Algumas interfaces interativas foram surgindo, tais como: interfaces de voz, telas sensíveis ao toque, reconhecimento de movimentos, entre vários outros. Observa-se também que tais inovações somente foram possíveis graças à evolução dos computadores.

O modelo ideal de interatividade é quando o usuário não mais percebe a presença da tecnologia, que estará trabalhando a seu favor de uma forma invisível e por toda a parte. Com base nisto, temos a Realidade Aumentada e a Realidade Virtual, que são tratadas como interfaces computacionais avançadas. Este trabalho busca mostrar mais um exemplo para a introdução de tal tecnologia na sociedade.

Passando pelos comandos de chaves e lâmpadas dos primeiros computadores, até os sistemas operacionais (SO) modernos, vemos cada vez mais o advento de novas técnicas para tornar mais fácil tal interação. Apesar das várias funções dos SO modernos, ainda ficamos presos a uma tela e a representações de ícones e *menus*. A integração da tecnologia no nosso dia-a-dia trouxe uma busca por uma relação mais natural e amigável do sistema com o

usuário. Essa busca levou ao desenvolvimento de diversos sistemas interativos, entre estes, a Realidade Virtual.

Atualmente podemos ver o propósito dessa tecnologia em prática em vários lugares, muitas vezes sem percebê-la: ao assistir um jogo de futebol, quando a TV mostra um tira-teima de um impedimento, traçando uma linha para confirmar um impedimento ou utilizando computação gráfica para reproduzir a posição dos jogadores; em vários programas com interação entre cenário e apresentadores como jornais (apresentando resumos virtuais no cenário enquanto o jornalista dá a notícia) e num caso mais famoso, no programa Fantástico da Rede Globo, onde os apresentadores interagem com o cenário; vemos também seu uso em campanhas publicitárias como as do Itaú e do Vectra que foram amplamente divulgadas em comerciais na TV.

1.1. Justificativa

A forma mais comum de se conhecer um local é visitá-lo. Partindo desse princípio, buscou-se aplicar as funcionalidades de Realidade Virtual e Aumentada para criar uma nova maneira de se conhecer um local.

Pensando na praticidade e funcionalidade para realizar as tarefas diárias, minimizando dificuldades e tempo, foi criado utilizando métodos de Realidade Aumentada (RA) a visualização ampla de ambientes imobiliários tornando mais objetiva, por exemplo, a escolha de um imóvel antes de ir ao seu local para visitá-lo, tendo uma perfeita visualização através de um ambiente virtual disponível na *web*. Ou ainda aos pais e filhos, conhecer ambientes escolares, hotéis, pontos turísticos, enfim, qualquer objeto de pesquisa que se tenham interesses específicos.

Foi buscado um meio simples e inovador para a exibição de diversos ambientes, de forma que o usuário da aplicação pudesse acessar de qualquer local, e dispor de poucos recursos. Tal forma de exibição de ambientes foca a usabilidade por meio do usuário, e a riqueza de detalhes possíveis de serem exibidos em um clipe 3D.

Objetiva-se proporcionar ao usuário uma sensação nova, através de recursos simples: uma webcam, um cubo montável e acesso à internet. Essa simplicidade dos recursos necessários torna a aplicação acessível à uma parcela maior da população, tornando este um meio mais eficiente de marketing.

Partindo da motivação pelo fascínio da interatividade proporcionada pelo uso da RA, pretende-se também despertar o interesse de novos usuários pela informática.

1.2. Objetivos

1.2.1. Objetivo Geral

Criar um moderno e objetivo sistema de marketing imobiliário motivado pela eficiência e objetividade proporcionadas por um sistema de Realidade Aumentada.

O sistema irá contribuir simplificando e popularizando a visualização de diversos ambientes e lugares.

1.2.2. Objetivos Específicos

Estudar a os seguintes temas:

- Teoria da Realidade Aumentada;
- Desenvolvimento Flash;
- Design 3-D;
- A biblioteca de Realidade Aumentada FLARToolKit;
- Interatividade.

Analisar trabalhos afins, buscando os melhores meios de desenvolvimento.

1.3. Estrutura da Monografia

O trabalho está dividido em quatro capítulos, sendo o capítulo um a introdução, que contém a apresentação dos objetivos gerais e específicos, além da justificativa e metodologia do trabalho. O capítulo dois apresenta o referencial teórico, onde são expostos os conceitos básicos para o entendimento do trabalho. No capítulo três são apresentados os resultados dos

testes e a implementação do trabalho. O capítulo quatro apresenta as considerações finais por meio da conclusão e sugestão para trabalhos futuros.

1.4. Metodologia

O desenvolvimento do sistema proposto foi fundamentado no uso da biblioteca Flash de Realidade Aumentada FLARToolkit combinada à biblioteca de renderização 3D Papervision3D. O uso dessa biblioteca possibilita a execução da aplicação diretamente do navegador de internet. É também portátil quanto à plataforma, sendo possível a sua execução em diversas arquiteturas e sistemas operacionais, desde que este possua instalado o *plugin* do FlashPlayer.

2. REFERENCIAL TEÓRICO

2.1. Realidade Virtual

A Realidade Aumentada faz parte dos estudos de Realidade Virtual, portanto, este projeto explicará primeiro o que é Realidade Virtual para depois tratar de Realidade Aumentada. Realidade Virtual é uma simulação do mundo real gerada por computador que propicia ao usuário do sistema uma interface avançada para execução de tarefas em tempo real no computador, oferecendo ao usuário a possibilidade de interagir com o sistema com vários recursos como visualização e movimentação, e também utilizando os sentidos, como visão, tato, audição e olfato.

De acordo com Pimentel (1995), a Realidade Virtual (RV) é o uso de alta tecnologia para convencer o usuário de que ele está em outra realidade, promovendo completamente o seu envolvimento.

Os primeiros relatos de seu uso datam do período pós 2ª Guerra Mundial, com o desenvolvimento dos primeiros simuladores de voo da Força Aérea dos Estados Unidos. Anos depois, em 1948, o cientista Morton Heilig desenvolveu uma máquina chamada Sensorama, que simulava a pilotagem de uma motocicleta. O Sensorama teve seu primeiro protótipo construído em 1962.



Figura 1 – Sensorama

Fonte: <http://thirddimensionknowledge.blogspot.com/2011/01/father-of-virtual-reality.html>

Devido ao desenvolvimento dos computadores e à popularização da tecnologia e, o estudo da Realidade Virtual foi ampliado, possibilitando seu uso em diversas áreas, tais como:

- Saúde;
- Entretenimento;
- Publicidade;
- Arquitetura;
- Educação;
- Treinamentos;
- Militar.

2.2. Ambiente Virtual

Ambiente virtual é a representação virtual de um determinado espaço físico reproduzindo fielmente as suas medidas de altura, largura e profundidade. Tem como princípio proporcionar ao usuário a sensação de fazer parte deste, utilizando os elementos disponíveis para sua interação.

Segundo Kirner (2003) a informação pode facilmente ser representada em mundos virtuais, principalmente porque seu formato visual é do modo coloquial, com o qual o usuário está acostumado no seu dia a dia. Através dos elementos existentes dentro do mundo virtual, e a sua representação no mundo real, o usuário sente-se à vontade em interagir com o ambiente.

2.3. Realidade Misturada

De acordo com Kirner (2003) a definição mais comum de Realidade Aumentada é a sobreposição de objetos virtuais tridimensionais, gerados por computador, com um ambiente real, por meio de algum dispositivo tecnológico. Porém esta é uma definição bastante genérica que só é esclarecida ao estudar a Realidade Misturada.

A Realidade Misturada apresenta duas modalidades: Realidade Aumentada e Virtualidade Aumentada. [KIRNER, C.; TORI, R. (2004)], de acordo com a imagem abaixo.



Figura 2- Diagrama de realidade/virtualidade contínua.

Fonte: (Kato, 2000)

2.4. Realidade Aumentada

Qual é a metáfora para o computador do futuro? O agente inteligente? A televisão multimídia? Um mundo gráfico tridimensional? O computador ubíquo, comandado por voz, da série Jornada nas Estrelas? A interface do Desktop remodelada? A máquina que magicamente realiza os desejos? Eu penso que a resposta certa seria “nenhuma das opções acima”, porque na minha opinião todos esses conceitos compartilham uma mesma falha básica: eles tornam o computador visível. Uma boa ferramenta é aquela invisível. Por invisível, quero dizer que a ferramenta não interfere na nossa consciência: você foca na tarefa e não na ferramenta. (Mark Weiser¹, 1991)

De acordo com Azuma (1997), Realidade Aumentada é uma linha de pesquisa da computação que visa integrar o mundo real com elementos virtuais ou dados gerados pelo computador. Os exemplos de sua aplicação mais vistos na atualidade são em jogos e campanhas publicitárias. A figura 3 mostra uma das primeiras campanhas a ter visibilidade: um jogo onde ao mostrar o marcador para a câmera, o usuário faz um *test drive* online, dirigindo o Vectra GT por uma pista virtual, e outra, mais recente, que exhibe os novos produtos e serviços do banco Itaú.

¹ Mark Weiser, ex-chefe cientista da Xerox PARC, considerado o pai da Computação Ubíqua e da Tecnologia Calma. (1991)



Figura 3 - Exemplos de uso da Realidade Aumentada na publicidade

Fontes: http://www.messa.com.br/eric/ecode/uploaded_imagens/vectra-760460.jpg e
<http://cricketdesign.com.br/blog/wp-content/uploads/2010/07/AFC-12021-011-Rv.jpg>

As pesquisas atuais buscam formas para seu uso na transmissão ao vivo de vídeos, onde são processados e adicionados gráficos do computador, e também seu uso em rastreamento de dados em movimento, reconhecimento mais avançado de marcadores e a construção de ambientes controlados que possuam qualquer número de controladores e atuadores.

A definição de Azuma (1997) é a mais aceita pelos pesquisadores: Realidade Aumentada é um ambiente que envolve tanto realidade virtual como elementos do mundo real, criando um ambiente misto em tempo real. Por exemplo, um usuário da RA pode utilizar óculos translúcidos, e através destes, ele poderia ver o mundo real, bem como imagens geradas por computador projetadas no mundo.

Também de acordo com Azuma(1997), um sistema de Realidade Aumentada seria aquele sistema capaz de:

- Combinar elementos reais com o ambiente real;
- Fornecer interatividade e processamento em tempo real;

- É desenvolvida em 3-D.

Considerando o sentido da visão, além de permitir que objetos visuais possam ser introduzidos em ambientes reais, a Realidade Aumentada também proporciona ao usuário o manuseio desses objetos com as próprias mãos, possibilitando uma interação natural e atrativa com o ambiente (Zorzal, 2009 apud Billinghamurst, 2001; Santin,2004;Zhou,2004).

Atualmente vemos aplicações de Realidade Aumentada em várias áreas: jogos, publicidade, educação, bioengenharia, física, geologia, entre outras.

2.5. Diferenças entre Realidade Virtual e Realidade Aumentada

O surgimento da realidade virtual trouxe vários avanços com relação à interatividade do usuário com o sistema. Com a evolução da tecnologia, foi possível o uso de computação gráfica em tempo real. Apesar destas vantagens, a Realidade Virtual se tornou um pouco incômoda, por ser necessário o uso de equipamentos como capacete, luvas, óculos 3-D, e muitas vezes também treinamento. Fatos que afastaram a tecnologia do público comum.

A evolução que permitiu o avanço dos estudos da Realidade Virtual também permitiu o surgimento da Realidade Aumentada, que implementou um novo conceito, onde o ambiente real é aumentado com objetos virtuais, utilizando algum dispositivo tecnológico. No início dos anos 2000, houve uma maior busca pelo desenvolvimento de aplicações que oferecessem um melhor custo-benefício. Foi então que o interesse pela Realidade Aumentada começou a ganhar força, o que a torna a tecnologia mais atraente para o desenvolvimento de interfaces para o público geral.

Diferente da Realidade Virtual, não há a necessidade de equipamentos caros e complexos, é flexível quanto às plataformas, oferece grande mobilidade, e oferece uma interface mais intuitiva ao usuário, se tornando mais abrangente.

2.6. RIA – Rich Internet Application

Os sistemas de Realidade aumentada são considerados como aplicações RIA. De acordo com a Adobe, aplicações RIA, Aplicações Ricas para Internet, são aplicações que oferecem algum tipo de interação com o usuário e são executadas diretamente do navegador de Internet. São exemplos: animações, vídeos, jogos, entre outros. Sua execução independe do sistema operacional, sendo assim de grande flexibilidade de uso. A maior parte das aplicações RIA são desenvolvidas em Flash, Silverlight, e JavaScript. As aplicações RIA possuem foco no usuário, utilizando conceitos da web 2.0, com maior enfoque na interatividade com o usuário.

2.7. Interatividade

Sistemas são considerados interativos quando oferecem ao usuário a possibilidade de responder às várias entradas possíveis, modificando em tempo real o ambiente ou o objeto virtual apresentado, proporcionando a sensação de imersão no sistema.

Sistemas interativos são bastante utilizados em aplicações educacionais (ensino a distância, jogos educativos), jogos e em ações de marketing (vídeos, jogos, *banners*).

2.7.1. Interface

Interface é qualquer dispositivo real ou virtual que oferece interação entre dois ou mais itens. São exemplos de interface: a tela de um computador, uma maçaneta, entre outros.

Ao focar apenas em sistemas de computação, percebe-se uma grande evolução quanto às interfaces de acordo com a tabela 1. Houve uma adaptação das interfaces para se adequarem conforme a demanda dos usuários.

Geração	Tipo de Uso	Usuários	Interface de interação
1945	Cálculos de Mecânica e Elétrica	Os próprios inventores	Movimentos de chaves e cabos
1955	Máquina de calcular, Processador de informações.	Grupos especializados sem conhecimento computacional.	Linhas de comando
1965	Mecanização de atividades repetitivas e não criativas.	Grupos especializados sem conhecimento computacional	Menus hierárquicos e preenchimento de formulários
1985	Computador como uma ferramenta.	Profissionais e curiosos	Mouse, ícones e janelas.
1995	Computador como um aparelho eletrônico	Todas as pessoas	Equipamentos multimídia
Atualmente	Computador como uma central multimídia	Todas as pessoas, todas as classes sociais.	Telas sensíveis, reconhecimento de imagens, voz e gestos.

Tabela 1 – Interfaces de interação computacionais

Fonte: Adaptado de (VIEIRA DA ROCHA; BARANAUSKAS, 2003).

2.7.2. Interação Humano – Computador (IHC)

“A Interação Humano-Computador (IHC) pode ser definida como a disciplina relativa ao design, avaliação e implementação de sistemas computacionais interativos para uso humano e aos fenômenos que os cercam”, (VIEIRA DA ROCHA; BARANAUSKAS, 2003).

De acordo com Norman (1998, apud VIEIRA DA ROCHA; BARANAUSKAS, 2003), um sistema ideal deve “esconder” a tecnologia do usuário, para que o usuário não perceba sua presença no SO. O objetivo é permitir que as pessoas façam suas atividades, com a tecnologia aumentando a sua produtividade por ser algo invisível ao usuário. O usuário deve ser capaz apenas de aprender a tarefa, não a tecnologia. As atividades do usuário devem ser simples, versáteis e prazerosas.

2.8. Adobe Flash

O Adobe Flash é uma ferramenta para o desenvolvimento de animações interativas.

As animações por ele geradas são executadas pelo navegador *Web* (Google Chrome, Mozilla Firefox, Internet Explorer, entre outros) através do Adobe Flash Player, uma máquina virtual capaz de executar as aplicações em Flash.

Percebe-se seu uso em: *Banners* de sites, *Charges* animadas, *Jogos via Web*, entre outros.

Atualmente está na versão CS5.5.

2.9. FlashDevelop

O FlashDevelop é uma ferramenta de desenvolvimento livre e de código aberto que oferece suporte para o ActionScript versões 2 e 3. Apesar de ser uma ferramenta de desenvolvimento Flash, não possui uma interface para desenvolvimento gráfico, sendo seus aplicativos desenvolvidos apenas por meio de código.

Para o desenvolvimento deste projeto, foi utilizada a versão 3.3.1.

2.10. ActionScript3

É uma linguagem de programação orientada a objetos. Foi desenvolvida com base em ECMAScript e seu uso é direcionado para o desenvolvimentos de RIA's (Rich Internet Applications - Aplicações Ricas de Internet).

Suas aplicações podem ser executadas em qualquer Sistema Operacional, por meio de uma Máquina Virtual.

2.11. ARToolkit

É uma biblioteca desenvolvida em C++ para o desenvolvimento de aplicações de Realidade Aumentada. O ARToolkit é uma biblioteca livre para usar em propósitos não comerciais e é distribuído com código aberto sob licença GPL.

É capaz de executar em diversas plataformas, entre elas: SGI Irix, PC Linux, PC Windows 95/98/NT/2000/XP/Vista e Mac OS X.

O ARToolKit foi desenvolvido pelo Dr. Hirozaku Kato da Universidade de Osaka, no Japão e apoiado pelo Human Interface Technology Laboratory da Universidade de Washington e de Canterbury, na Nova Zelândia e também pelo laboratório de multimídia do MIT. Usa técnicas de visão computacional para calcular a posição no espaço real da câmera e sua orientação em relação aos cartões marcadores, o que permite ao desenvolvedor a sobreposição dos objetos virtuais pelos cartões. O pacote inclui bibliotecas de rastreamento e disponibiliza o código fonte completo, tornando possível o transporte do código para diversas plataformas ou adaptá-los para resolver as especificidades de suas aplicações.

O ARToolkit usa técnicas de visão computacional para o cálculo da posição real da câmera e a orientação dos marcadores.

Para isso, o marcador é capturado pela câmera e transformado em binário, a partir do valor do limiar; procura-se então nesta imagem por regiões quadradas ou retangulares, e as compara com um padrão existente na base de dados. Caso a imagem relativa à região quadrada seja reconhecida como um marcador, a posição da imagem é calculada. Uma matriz 3X4 é preenchida com as coordenadas relativas ao marcador. Com base nessa matriz, define-se o conjunto das coordenadas virtuais correspondentes ao objeto virtual a ser posicionado no marcador.

Para calcular as coordenadas virtuais da câmera e desenhar as imagens virtuais é utilizada a API OpenGL.

A imagem a seguir ilustra o funcionamento da biblioteca:



Figura 4 - Diagrama de funcionamento do ARToolkit

Fonte: (Kirner, 2004)

2.12. FLARToolKit

O FLARToolKit [Koyama, 2010] é uma biblioteca de Realidade Aumentada em ActionScript traduzida por Saquoosha em 2008 a partir do NyARToolKit, que é escrita em Java. Esta linguagem pode ser executada por qualquer *browser* com o *plugin* do Adobe Flash Player. É também uma plataforma bastante flexível, possibilitando o desenvolvimento de aplicações para diversas plataformas, como demonstra a Tabela 1. Basicamente, é possível desenvolver aplicações com o FLARToolKit para qualquer computador ou dispositivo móvel que ofereça suporte ao Flash. Seus aplicativos podem ser executados localmente ou como cliente/servidor.

Tecnologias de Realidade Aumentada	Plataformas		
	Desktop	Web	Dispositivos Móveis
ARStudio	Sim	Não	Não
ARTag	Sim	Não	Não
ARToolKit	Sim	Não	Não
ARToolKitPlus	Não	Não	Sim
DART	Sim	Não	Não
FLARToolKit	Sim	Sim	Sim
JARToolKit	Sim	Não	Não

Tabela 2 – Comparação entre as bibliotecas de Realidade Aumentada

Segundo Kirner, 2011, sua plataforma livre, aliada ao fato de vários profissionais de publicidade utilizarem o Flash em seus trabalhos, o fez ganhar visibilidade e popularidade em aplicações de Realidade Aumentada.

A biblioteca FLARToolKit possui diversas funções interativas, possibilitando seu uso em aplicações com ações variadas. Como exemplos: comandos de voz, colisão entre objetos virtuais, oclusão de marcadores, cálculo de distância entre diversos marcadores, orientação espacial dos marcadores, entre outras funcionalidades.

Segundo Koyama (2010) seu funcionamento segue etapas bem parecidas com as do ARToolkit:

- I. Captura da imagem da câmera – É realizada durante toda a aplicação, considerando os parâmetros de câmera e marcadores já definidos.



Figura 5 – Imagem capturada pela câmera

Fonte: (Siscoutto e Filho apud KATO, 2011)

- II. Criação do mapa de bits da imagem limiarizada – A partir da imagem capturada, a imagem é binarizada, para facilitar a detecção de possíveis marcadores.



Figura 6 - Imagem limiarizada

Fonte: (Siscoutto e Filho apud KATO, 2011)

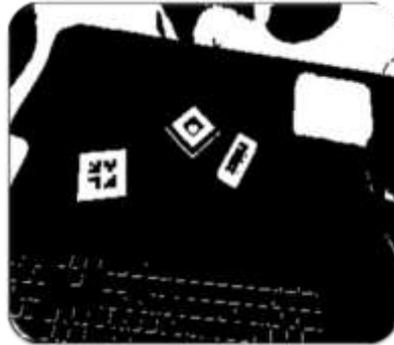


Figura 7 - Imagem binarizada

Fonte: (Siscoutto e Filho apud KATO, 2011)

III. Detecção dos quadrados – É feita uma busca por quadrados (possíveis marcadores) rastreando a imagem capturada.



Figura 8 - Marcadores encontrados pelo FLARToolkit

Fonte: (Siscoutto e Filho apud KATO, 2011)

Na imagem acima, foram encontrados quatro possíveis marcadores. Como forma de otimização do desempenho, é tomado como padrão para a análise de identificação, o FLARToolkit utilizar apenas 50% da área do possível marcador para analisá-lo com os marcadores já cadastrados. Dependendo da complexidade entre marcadores exigida pela aplicação, o desenvolvedor pode alterar a porcentagem da área a ser rastreada alterando os valores dos parâmetros *i_markerPercentWidth* e *i_markerPercentHeight* na função *FLARCode()* – em *org.libspark.flartoolkit.core.FLARCode*.

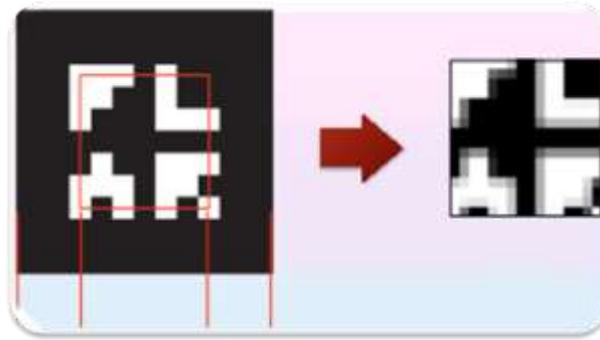


Figura 9 - Área do marcador utilizada para comparação

Fonte: (Siscoutto e Filho apud KATO, 2011)

IV. Identificação e comparação dos marcadores – Após o reconhecimento de possíveis marcadores pelo sistema, estes são comparados com os marcadores já registrados. Para esta comparação, cada marcador registrado é comparado com cada marcador identificado, de forma que a comparação se realize através da técnica de homografia, onde se faz uma comparação a cada rotação de 90° do marcador registrado com um marcador identificado, gerando um resultado de 0 a 1 (Siscoutto e Filho, 2011). A proximidade do valor encontrado a 1 irá definir se o marcador encontrado é o mesmo cadastrado pela aplicação. Como um segundo parâmetro de comparação, é também definido um grau de confiança, onde o valor encontrado deve sempre ser superior a este. Como definição geral, costuma-se utilizar o grau de confiança superior a 0,7, ou seja, o marcador encontrado tem que possuir no mínimo 70% de semelhança com o marcador cadastrado. Este grau de confiança é importante, pois a imagem captada pela câmera pode sofrer deformações de acordo com a luminosidade, ângulo de captura, tamanho, entre outros fatores. Portanto, o grau de confiança deve ser um valor suficiente para desprezar pequenas deformidades do marcador correto, e não reconhecer marcadores incorretos.

				
	0.38	0.29	-0.15	-0.11
	0.86	0.20	-0.01	-0.14
	0.27	-0.03	0.03	-0.14
	0.13	0.16	-0.08	-0.01

Figura 10 - Tabela de comparação dos marcadores identificados

Fonte: (Siscoutto e Filho apud KATO, 2011)

O aplicativo reconhece como marcador a imagem cujo valor encontrado seja o maior valor maior que o grau de confiança.



Figura 11 - Destaque para o marcador reconhecido pelo FLARToolkit

Fonte: (Siscoutto e Filho apud KATO, 2011)

V. Cálculo da matriz de transformação – Após a identificação dos marcadores, o aplicativo determina o posicionamento do objeto virtual a ser inserido. É calculada uma matriz de transformação a partir de um sistema de coordenadas da câmera. O FLARToolkit utiliza para realizar o cálculo com um ponto 3D no marcador identificado, uma matriz 3X4, denominada Matriz Transformação. O resultado deste cálculo determina o ponto no sistema de coordenadas da câmera em que o objeto virtual será situado (Siscoutto e Filho apud Kirner, 2011).

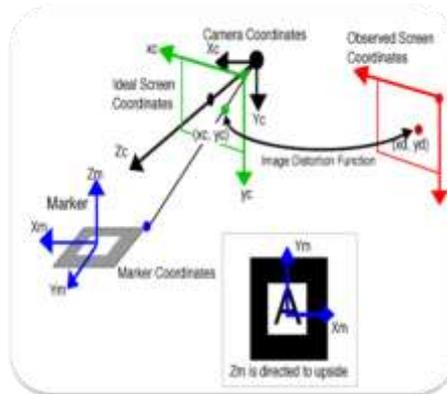


Figura 12 - Relacionamento entre as coordenadas do marcador e da câmera

Fonte: (Siscoutto e Filho apud KATO, 2011)

O FLARToolKit utiliza os parâmetros de câmera para realizar a correção de distorções, assim como no ARToolKit. Segundo Cabana (2010), O arquivo câmera_para.dat encontrado na biblioteca do FLARToolKit representa estes parâmetros, sendo considerado o cerne do FLARToolKit, que muito provavelmente não precisará ser alterado, a não ser em casos específicos e em projetos mais avançados.

VI. Renderização dos objetos virtuais – Com o auxílio de uma *engine* 3D (neste projeto será utilizado o Papervision3D), os objetos serão renderizados na tela a partir das informações de posicionamento e qual foi o objeto selecionado.



Figura 13 - Objeto virtual integrado ao ambiente real

Fonte: (Siscoutto e Filho apud KATO, 2011)

2.13. Ferramentas 3D

É necessário o uso de ferramentas 3D para a criação e a renderização dos modelos a serem exibidos. No projeto foi dada ênfase no uso de ferramentas gratuitas, como o Google Sketchup e o Blender3D para a edição dos modelos, e o Papervision3D para a renderização dos objetos 3D na cena.

2.13.1. Papervision3D

Segundo Saqoosha (2010), o FLARToolKit apenas calcula a orientação dos marcadores, definindo a posição dos objetos virtuais e outras ações como movimentação e escala, mas não renderiza os objetos 3D da cena. Sendo assim, se faz necessário o uso de uma biblioteca específica para objetos 3D. Uma das bibliotecas disponíveis é o Papervision3D, que é uma biblioteca em ActionScript 3.0, e faz uso da API 3D do Flash. para gerar animações e interações em 3D. O FLARToolKit utiliza o Papervision3D para a renderização dos objetos 3D de diversos formatos.

A tabela abaixo detalha as classes e formatos suportados pelo Papervision3D.

Classe	Descrição
Ase	Permite carregar objetos exportados do 3DSMax
Collada	Permite carregar cenas Collada
DAE	Arquivo convertido para o formato COLLADA 1.4.1
KMZ	Modelo para o Google Earth
Max3DS	Arquivo 3DS
MD2	Carrega um arquivo do Quake 2 MD2 com animação
Sketchup	Modelo do Google Sketchup
Sketchup Collada	Modelo do Google Sketchup para Collada

Tabela 3 - Classes e formatos suportados pelo Papervision3D

2.13.2. Google SketchUp

É uma ferramenta de desenvolvimento e modelagem em 3-D. É um produto do Google e pode ser baixado gratuitamente em sua versão mais simples.

É amplamente utilizado em projetos de Arquitetura, design, engenharia, construção, educação e entretenimento digital.

2.13.3. Blender3D

É uma ferramenta de código aberto e multiplataforma usada para o desenvolvimento e modelagem em 3-D. Com esta ferramenta é possível também a criação de aplicações 3D interativas, como apresentações e jogos.

É uma ferramenta que vem ganhando bastante projeção na comunidade de desenvolvedores 3D. Já foi recomendada como ferramenta oficial do *Peugeot Design Contest*, concurso de design de carros da montadora Peugeot. Atualmente está na versão 2.60a

2.14. Códigos Interativos

No nosso dia-a-dia temos contato com alguns códigos capazes de armazenar diversos tipos de informações. São eles: os códigos de barras unidimensionais, utilizados principalmente no comércio, para armazenar dados dos produtos, e os códigos bidimensionais, mais poderosos, capazes de armazenar pequenos textos, *links* de internet, e também definir ações tais como realizar uma ligação ou enviar um SMS.

Estes códigos são considerados os precursores da Realidade Aumentada, pois ao serem interpretados são capazes de inserir informações num plano virtual.

2.14.1. Códigos de barras

Os Códigos de barras são definidos por um conjunto de barras paralelas verticais, O código de barra é uma representação gráfica de barras verticais paralelas, claras ou escuras, que pode ser lida de forma óptica e revela os dados do produto sob o qual está aplicado (FITTIPALDI, 2011 apud ARAUJO, 2009 e SEVERO FILHO, 2006).

Os códigos de barras unidimensionais são capazes de armazenar no máximo 20 dígitos.



Figura 14 – Exemplo de código de barras

Fonte: <http://gostin.com.br/blog/wp-content/uploads/2010/04/codbarra-500por300.jpg>

2.14.2. QR Code

De acordo com o site Oficina da Net, QR Code é um código de barras bidimensional criado pela empresa japonesa Denso-Wave em 1994. QR significa "quick response" (resposta rápida), nome dado devido à sua capacidade de ser interpretado rapidamente. QR Codes são bastante utilizados no Japão, porém seu uso ainda é pouco explorado no Brasil.



Figura 15 – Exemplo de um QR Code contendo um texto

Os QR Codes podem ser facilmente gerados através de diversos sites, entre eles o site “<http://qrcode.kaywa.com>”. De acordo com o site “qrcode.com”, mantido pela Denso-Wave, o QR Code contém informações na vertical e horizontal, enquanto um código de barras contém informações em apenas uma direção. QR Code tem uma capacidade de armazenamento de dados consideravelmente maior do que um código de barras, como exemplificam a figura 16 e a tabela 4.



Figura 16 – Direções das informações dos códigos

Tipo de dados	Capacidade de armazenamento
Numérico	7089 caracteres
Alphanumeric	4296 caracteres
Binário – 8 bits	2953 bits
Kanji – Alfabeto japonês	1817 caracteres

Tabela 4 – Capacidades de armazenamento de um QR Code

Atualmente, percebe-se seu uso em campanhas publicitárias, em revistas, para acesso a conteúdos exclusivos da web, e também em informações de diversos produtos.

Tornou-se bastante popular devido à grande presença de celulares com recurso de câmera.



Figura 17 – Leitura de um QR Code por um Celular

Disponível em: <http://info.abril.com.br/professional/img/qr-code17042009.jpg>

2.15. Marcador

A representação do objeto gerado pelo Papervision3D é feita por meio do reconhecimento do marcador correto.

Neste projeto foi definida a utilização de marcadores fiduciais. Estes foram um dos primeiros padrões a serem utilizados. Consiste de um quadrado de bordas pretas com um padrão central capaz de torná-lo único e distinguível de outros marcadores. Não existem restrições quanto à sua imagem central, ficando esta a critério do desenvolvedor. A detecção deste marcador é feita pela comparação de trechos da imagem capturada com um banco de dados de marcadores até que seja encontrado um que respeite o grau de confiança definido.

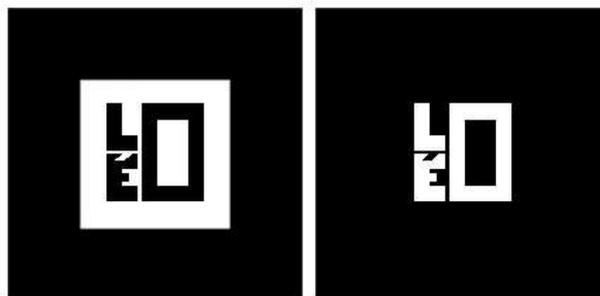


Figura 18 - Exemplos de marcadores fiduciais

Existem diversas ferramentas para geração de marcadores fiduciais. Entre estas, no site Marker Generator (<http://flash.tarotaro.org/ar/MGO2.swf>), onde pode-se criar um marcador através de uma imagem já impressa, ao mostrá-la para a webcam ou realizando um upload de uma imagem do arquivo. Suas opções de configuração:

- Mode Select – Define se a imagem será capturada pela câmera (Camera Mode) ou se será realizado um upload da imagem (load marker image).
- Marker Size – Define a área a ser utilizada para a identificação do marcador. Como dito antes, toma-se por padrão a utilização de 50% da área, um valor que pode ser alterado dependendo da complexidade da aplicação. Quanto menor a área de identificação, menor será o custo de processamento, e mais difícil será a identificação correta do posicionamento do marcador.
- Marker Segments – Define o tamanho e a qualidade do marcador. Quanto maior a qualidade, maior o arquivo .pat gerado e maior o custo de processamento, porém, a identificação será mais confiável.
- Get Pattern – Salva o marcador após o seu reconhecimento. Possui ainda uma janela com o nome Preview Marker onde é possível a visualização de como será o reconhecimento do marcador pelo Flash.

A figura 15 mostra o Marker Generator em funcionamento. Uma borda vermelha é traçada em volta do marcador reconhecido pelo programa.



Figura 19 - Marker Generator em funcionamento

Para a geração deste marcador, deve-se atentar para a escolha de um marcador que seja de fácil reconhecimento e não possibilite interpretações ambíguas. Um bom modelo de marcador deve apresentar uma diferença a cada rotação de 90° no eixo “z”, facilitando o trabalho do FLARToolKit na orientação x, y e z do marcador para o posicionamento de um objeto virtual [Cabana, 2010].

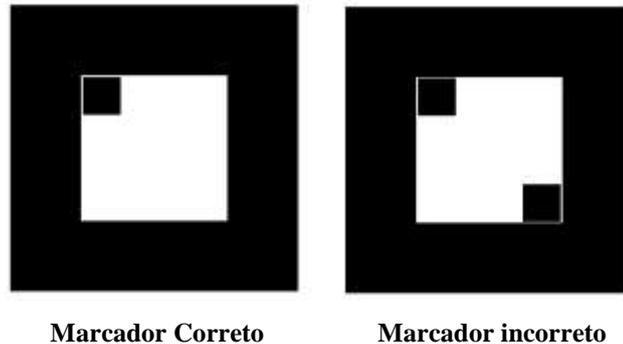


Figura 20 – Exemplos de marcadores corretos e incorretos

É necessária atenção quanto ao modo de interpretação das posições x, y e z representadas pela imagem do marcador em um plano 3D. A coordenada “x” é representada pelo deslocamento em pé (de cabeça para cima) do marcador. A coordenada “y” é o deslocamento horizontal do marcador e a coordenada “z” é dada pela aproximação/afastamento do marcador da webcam.

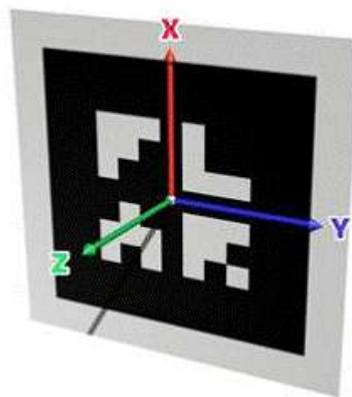


Figura 21 – Coordenadas do marcador

Fonte: <http://saqoosha.net/en/flartoolkit/start-up-guide/>

3. DESENVOLVIMENTO

No desenvolvimento do projeto foi utilizada a biblioteca Flash para Realidade Aumentada FLARToolKit, e o método de marcadores individuais, Tendo como interface de interação com a aplicação um cubo montável com marcadores diferentes em cada face, cada um referenciando a um ambiente diferente e com a função de controlar as ações do ambiente virtual (Rotação, inclinação e zoom).

3.1. Interface de interação

Visando tornar a manipulação da interface de interação mais fácil e prática, foi definida a impressão dos marcadores em um cubo. Tal modelo visa também atrair a atenção de crianças para a aplicação.

As figuras 21 e 22 exemplificam o cubo, desmontado e após montado.

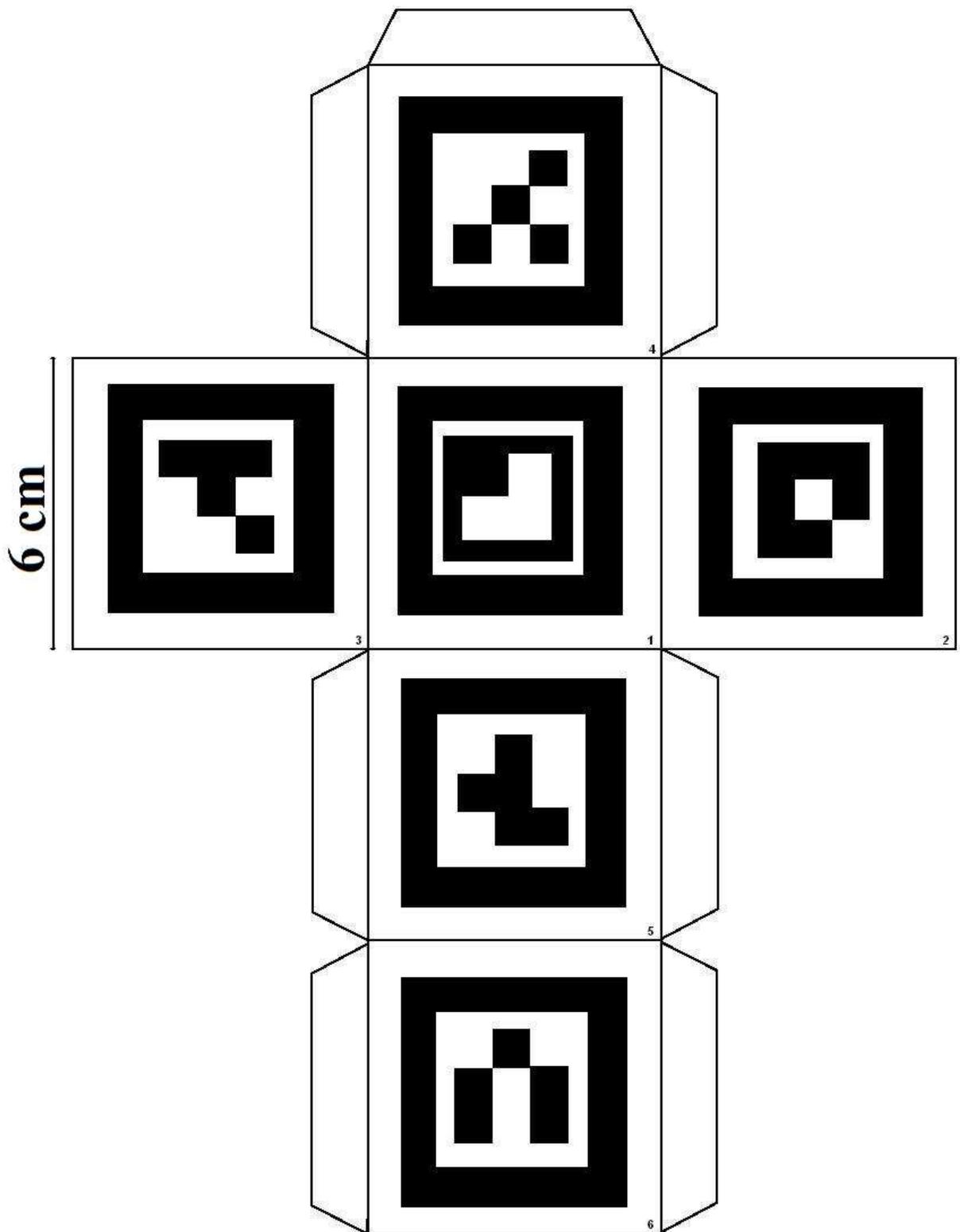


Figura 22 - Cubo montável com marcadores representando os diversos ambientes



Figura 23 - Cubo montado

Para o controle das ações do ambiente virtual foi utilizada a técnica do cálculo da posição dos marcadores, onde o FLARToolKit executa o processamento das imagens, coletando sua posição e orientação num ambiente 3D.

O sistema apresenta uma sequencia lógica bem definida, onde após a confirmação do usuário a câmera é inicializada e o sistema detecta a presença do marcador no frame de vídeo captado, exibindo o ambiente correspondente.

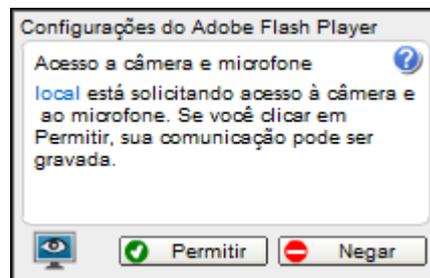


Figura 24 – Tela de confirmação de acesso à câmera

Após isso, detecta a presença e a orientação do marcador de controle no frame de vídeo captado e executa as ações de acordo com o posicionamento e orientação do marcador. O fluxo é exemplificado na figura abaixo.

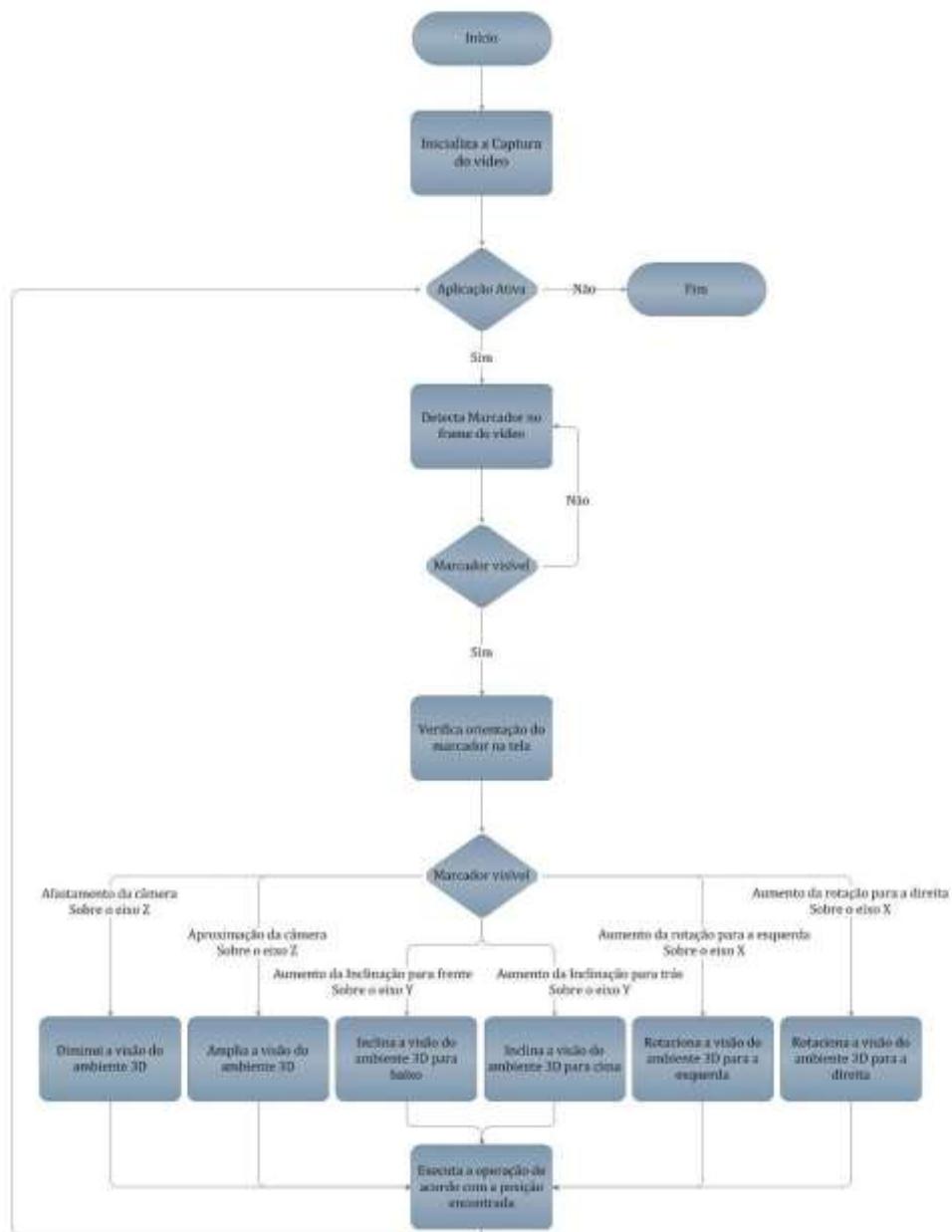


Figura 25 - Diagrama do fluxo de funcionamento do sistema

3.2. Ferramentas utilizadas

Para o início do desenvolvimento foi necessário o download e instalação das seguintes ferramentas:

Nome	Descrição	Download
FLARToolKit 2.5.4	Biblioteca do FLARToolKit	http://www.libspark.org/wiki/saqoosha/FLARToolKit/downloadlist
Flex SDK 4.1	Kit de desenvolvimento para compilação dos aplicativos	http://www.adobe.com/cfusion/entitlement/index.cfm?e=flex4sdk
FlashDevelop 3.3.1	IDE para desenvolvimento das aplicações	http://www.flashdevelop.org/downloads/releases/FlashDevelop-3.3.1-RTM.exe
Java Runtime 6	Java Virtual Machine necessária para execução do FlashDevelop	http://www.java.com/pt_BR/download/

Tabela 5 – Ferramentas instaladas

Após a instalação das ferramentas, foi criado um novo projeto no FlashDevelop e foi criado em sua pasta lib uma pasta para cada biblioteca necessária para o projeto. Em cada pasta, foi copiada a pasta src das respectivas bibliotecas, conforme a imagem abaixo.

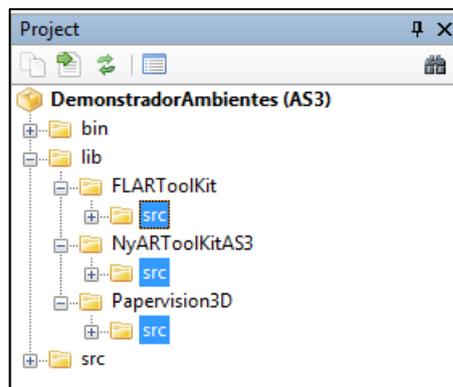


Figura 26 - Estrutura das pastas do projeto no FlashDevelop

3.3. Classes

Para o desenvolvimento da aplicação, é necessário o uso de classes básicas do Flash, do FLARToolKit e do Papervision3D, as quais serão representadas por atributos do projeto.

São as classes básicas:

PACOTE	CLASSE	DESCRIÇÃO
Flash	URLLoader	Carrega dados binários ou texto através de uma URL informada. Utilizado para carregar os parâmetros da câmera e marcador em uma aplicação de RA
Flash	Stage	Palco de exibição principal de conteúdo gráfico
Flash	Camera	Webcam
Flash	Bitmap	Imagem capturada em mapa de <i>bits</i>
Flash	Video	Video para exibição do conteúdo
Flash	Sprite	Nó para exibição de gráficos/conteúdo 3D
FLARToolKit	FLARRasterThresholdAnalyzer_SlidePTile	Analisa o nível de limiarização da imagem
FLARToolKit	FLARBaseNode	Container para o modelo 3D
FLARToolKit	FLARCode	Marcador detectado
FLARToolKit	FLARTransMatResult	Matriz de transformação
FLARToolKit	FLARSingleMarkerDetector	Detector de um único marcador
FLARToolKit	FLARParam	Parâmetros da câmera
FLARToolKit	FLARRgbRaster_BitmapData	Rasterizador RGB - utilizado para transformar a imagem rgb para mapa de bits
Papervision3D	Viewport3D	Container para todo o conteúdo (modelo, marcador, camera3D)
Papervision3D	Camera3D	Câmera virtual
Papervision3D	Scene3D	Cena virtual
Papervision3D	DAE	Modelo virtual do tipo COLLADA 1.4.1
Papervision3D	LazyRenderEngine	Renderizador Papervision3D

Tabela 6 – Classes básicas do FLARToolkit

3.4. Padrões Utilizados

Para o desenvolvimento do trabalho, foram definidos alguns parâmetros relevantes: tamanho do vídeo a ser exibido, tamanho, porcentagem de reconhecimento e o número de

segmentos do padrão dos marcadores, e também número de *frames* a serem exibidos por segundo.

Para a definição do tamanho do vídeo, foram considerados: resolução mais usual dos monitores, tamanho das barras de ferramentas dos navegadores e do SO, e a capacidade de visualização tanto em monitores widescreen quanto em monitores comuns. Com isso, o tamanho do vídeo foi definido em 640 *pixels* de largura por 480 *pixels* de altura.

Para os marcadores, foi definida a porcentagem de reconhecimento em 50%, seu tamanho em 6 (seis) centímetros, e o seu padrão dividido em 16 segmentos. Foi também definida a sua exibição com 16 *frames* por segundo. Tais medidas consideram o seu uso na web, exigindo menos processamento, não limitando assim o uso em computadores mais lentos.

```

45 public function FLARCode(i_width:int, i_height:int, i_markerPercentWidth:uint = 50, i_markerPercentHeight:uint = 50)
46 {
47     super(i_width, i_height);
48     this._markerPercentWidth = i_markerPercentWidth;
49     this._markerPercentHeight = i_markerPercentHeight;
50 }

```

Figura 27 - Função FLARCode com parâmetros padrões

3.5. Funções criadas

A leitura da posição do marcador é a base deste trabalho. A partir dessa leitura, serão definidas as interações como modelo 3D. Para o desenvolvimento dessa função, foi considerada a necessidade de se tratar os valores a cada *frame*.

O FLARToolkit já oferece a função *onFrameCapturado*, na classe ProjetoPapervision3D, que analisa o *frame* capturado e informa para a aplicação se há a presença de um código válido, com a variável booleana *isMarcadorDetectado*. Caso haja a presença de um marcador no *frame*, é calculada a matriz de transformação do marcador e seus valores são atribuídos ao modelo 3D. Para o cálculo do posicionamento do marcador, foram aproveitados os valores de X, Y e Z lidos pela função, e atribuídos às variáveis do tipo *number* “*valorX*”, “*valorY*” e “*valorZ*”, conforme a figura 28.

```

Main.as  ProjetoPapervision3D.as  InteracaoPapervision3D.as  FLARSingleMarkerDetector.as
340      private function onFrameCapturado(e:Event = null):void {
341
342          //imagem capturada pela câmera é apresentada na tela do usuário
343          capturaBitmap.bitmapData.draw(video);
344
345          //Informa à aplicação se o marcador foi detectado ou não.
346          isMarcadorDetectado = (detectorUnicoMarcador.detectMarkerLite(
347              rasterizadorRGB, limiarizacao) &&
348              (detectorUnicoMarcador.getConfidence() > 0.5));
349
350          //Se Positivo: a matriz de transformação é calculada e passada para a
351          //base do modelo virtual (get e setTransformMatrix)
352          if (isMarcadorDetectado) {
353              baseModelo.visible = true;
354              detectorUnicoMarcador.getTransformMatrix(matrizTransformacao);
355              baseModelo.setTransformMatrix(matrizTransformacao);
356
357              valorX = this.baseModelo.position.x;
358              valorY = this.baseModelo.position.y;
359              valorZ = this.baseModelo.position.z;
360              TextoResult();
361          }

```

Figura 28 - Função onFrameCapturado

Como forma de exemplificar o funcionamento do sistema, foi inserida na tela uma caixa de texto contendo os valores atualizados de X, Y e Z. Como tratado anteriormente, as variáveis que armazenam estes valores são do tipo *number*. Por isso, ao serem lidas antes de serem inicializadas com os valores válidos de posição, retornam NaN, *Not a Number*, conforme destaque na figura 29.

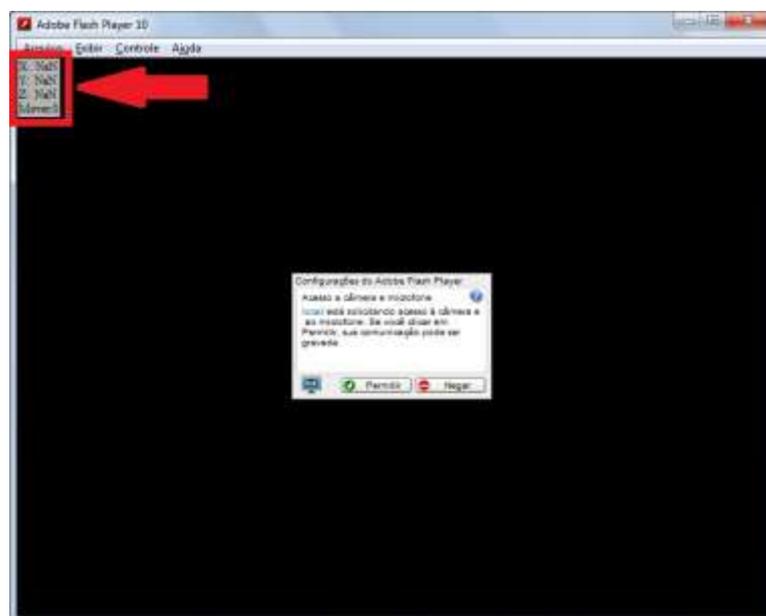


Figura 29 - Função onFrameCapturado

Para o cálculo dos valores de movimentação, foi criada a função *MovimentoXYZ* e determinado um intervalo de leitura dos valores de posição. A cada 15 *frames* a variável de controle de posição é atualizada e seu valor atual é comparado ao valor anterior. Foram

criadas variáveis separadas para as posições X, Y e Z. Para se atribuir o movimento, considera-se a direção em que a diferença entre os valores iniciais e finais tenha sido maior. Caso a maior movimentação reconhecida seja maior que o valor mínimo de movimentação, atribui-se à variável de controle um valor seguindo o padrão de acordo da tabela 7.

Valor retornado	Movimento
0	Não movimenta
1	Positivo em X
2	Negativo em X
3	Positivo em Y
4	Negativo em Y
5	Positivo em Z
6	Negativo em Z

Tabela 7 – Padrão de atribuição de movimento

```

ProjetoPapervision3D.as
169      public function MovimentoXYZ(): Number
170      {
171          var constante: Number = 15; // menor movimento para ser reconhecido
172          if (cont == 15)
173          {
174              cont = 0;
175              val2_X = val1_X;
176              val1_X = getValorX();
177              x_val = val2_X - val1_X;
178              val2_Y = val1_Y;
179              val1_Y = getValorY();
180              y_val = val2_Y - val1_Y;
181              val2_Z = val1_Z;
182              val1_Z = getValorZ();
183              z_val = val2_Z - val1_Z;
184
185              //Atribuir valores a moverX, moverY e moverZ
186              if (x_val < constante) moverX = 1; //mover x direita
187              if (x_val > constante) moverX = 2; //mover x esquerda
188              if (y_val < constante) moverY = 3; //mover y frente
189              if (y_val > constante) moverY = 4; //mover y trás
190              if (z_val < constante) moverZ = 5; //mover z frente
191              if (z_val > constante) moverZ = 6; //mover z trás
192
193              //Tornar os valores de movimentação positivos
194              if (x_val < 0) x_val *= -1;
195              if (y_val < 0) y_val *= -1;
196              if (z_val < 0) z_val *= -1;
197
198              //Comparar o maior movimento
199              if (x_val > y_val || x_val > z_val)
200                  mover = moverX;
201              else
202                  if (y_val > x_val || y_val > z_val)
203                      mover = moverY;
204                  else
205                      if (z_val > x_val || z_val > y_val)
206                          mover = moverZ;
207              }
208          }
209          else
210          {
211              cont = cont + 1;
212          }
213          return mover;
214      }

```

Figura 30 - Função MovimentoXYZ

Após a execução da função *MovimentoXYZ*, é chamada a função *VerificarAcoes* da classe *ProjetoPapervision3D* para atribuir o movimento ao modelo 3D.

```

ProjetoPapervision3D.as*
440     public function VerificarAcoes():void
441     {
442         var mov:ProjetoPapervision3D = new ProjetoPapervision3D();
443
444         switch (mover) {
445
446             case 1:
447                 baseModelo.getChildByName("modelo").rotationZ += 5;
448                 break;
449             case 2:
450                 baseModelo.getChildByName("modelo").rotationZ -= 5;
451                 break;
452             case 3:
453                 baseModelo.getChildByName("modelo").rotationY += 5;
454                 break;
455             case 4:
456                 baseModelo.getChildByName("modelo").rotationY -= 5;
457                 break;
458             case 5:
459                 //baseModelo.getChildByName("modelo").rotationX += 5;
460                 baseModelo.getChildByName("modelo").scale += 5;
461                 break;
462             case 6:
463                 //baseModelo.getChildByName("modelo").rotationX -= 5;
464                 baseModelo.getChildByName("modelo").scale -= 5;
465                 break;
466         }
467     }
468 }
469 }

```

Figura 31 - Função verificarAcoes

3.6. Testes

Quando é exibido um marcador válido, são exibidos, além do modelo 3D, os valores das posições de X,Y e Z no plano tridimensional e o movimento a ser realizado, conforme destaque na figura 32. A variável *mover* é atualizada de acordo com o movimento realizado pelo marcador. O modelo 3D é movimentado de acordo com os valores da variável *mover*.

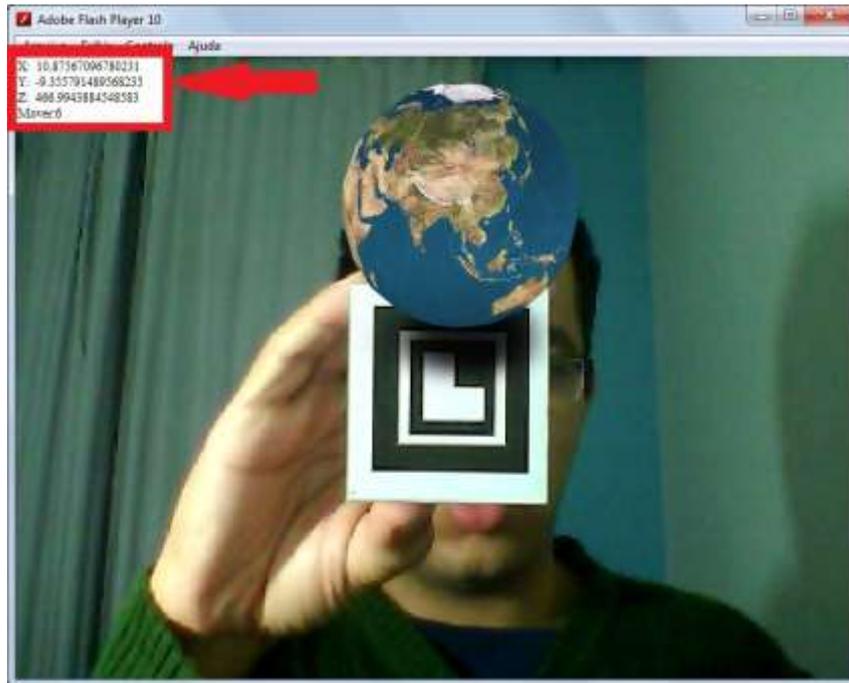


Figura 32 – Tela com os valores de posição e a ação a ser realizada

3.7. Aplicação Web

Foi criada uma página web para o usuário acessar a aplicação. A página pode ser acessada pelo endereço *web* “www.aumentando.xpg.com.br” ou pelo QRCode representado pela figura 33.



Figura 33 - QRCode para acesso à aplicação Web



Figura 34 - Aplicação Web

Neste site, são disponibilizadas para o usuário algumas informações sobre Realidade Aumentada, o demonstrador de ambientes e um arquivo PDF com o cubo para ser impresso e montado.

3.8. Testes finais

A imagem abaixo mostra a execução da aplicação com a exibição do Estádio Mineirão, de Belo Horizonte.



Figura 35 - Aplicação Web ativa exibindo o estádio Mineirão

4. CONSIDERAÇÕES FINAIS

4.1. Conclusão

Um Demonstrador Virtual de Ambientes via web é capaz de elevar a experiência final do usuário ao poder interagir com o ambiente sem a necessidade estar presente no mesmo, além da praticidade e mobilidade de acesso por meio da internet.

Apesar das facilidades da virtualidade e o acesso pela internet, a necessidade do uso de teclado e *mouse* dificultavam a interação do sistema com o usuário. A integração dos recursos de Realidade Aumentada elevou o nível de interação do usuário, trazendo o ambiente para perto dele e os controles em suas mãos, com movimentos simples e intuitivos.

No decorrer do desenvolvimento do projeto, houve algumas dificuldades, como na utilização de uma biblioteca para Realidade Aumentada em Java, o JARToolKit, que após vários erros encontrados, foi descoberto que era uma biblioteca descontinuada, portando, sem suporte disponível. Após esta descoberta, foi decidida a troca pela biblioteca Flash FLARToolKit. Outra dificuldade foi pelo uso da linguagem ActionScript3, para programação em Flash, que até então não a conhecia.

Como pontos positivos, podem ser salientados os conhecimentos adquiridos: em uma nova linguagem de programação, em uma nova área de pesquisa, a de Realidade Virtual, e na participação no SVR2011 – XVIII Simpósio de Realidade Virtual e Aumentada, em Uberlândia, onde adquiri grande parte do meu conhecimento atual sobre Realidade Virtual e Aumentada, sendo de grande valia no desenvolvimento final do projeto.

4.2. Trabalhos futuros

Como trabalhos futuros são propostas a implementação da aplicação para as plataformas Android e iOS (iPhone e iPad), e também o desenvolvimento em outras linguagens de programação.

É proposto também a implementação da funcionalidade de múltiplos marcadores, tornando possível, assim, realizar a troca dos ambientes 3D conforme o usuário trocar a face do cubo.

5. REFERENCIAS

1 up, Blog. Disponível em: <<http://www.1up.net.br/2011/01/flash-flex-ria-sorria/>>. Acesso em 15 nov. 2011.

Adobe. Disponível em: <<http://www.adobe.com/>>. Acesso em 25 out. 2011.

AISA. Disponível em: <<http://www.aisa.com.br/oquee.html>>. Acesso em 25 out. 2011.

AZUMA, R. T. **A Survey of Augmented Reality. Presence: Teleoperators And Virtual Environments**, vol.6, n. 4, Aug. 1997.

CARDOSO, A.; LAMOUNIER JR, - **Realidade Virtual: Uma Abordagem Prática.** Livro dos Minicursos do SVR2004, SBC, São Paulo, 2004.

CRICKET DESIGN, Blog. Disponível em: <<http://cricketdesign.com.br/blog/?p=673>>. Acesso em 15 nov. 2011.

C-LABS – Homepage. Disponível em: <<http://www.c-lab.de/jartoolkit/>>. Acesso em 15 nov. 2011.

Denso-Wave. Disponível em: <<http://www.qrcode.com>>. Acesso em 15 nov. 2011.

GABRIEL, Martha. **QR codes & Realidade Aumentada.** 2010. Disponível em: <<http://www.slideshare.net/marthagabriel/qr-codes-realidade-aumentada-por-martha-gabriel>>. Acesso em 15 nov. 2011.

Google Sketchup. Disponível em: <<http://nyatla.jp/nyartoolkit/wiki/index.php?NyARToolkit%20for%20Java.en>>. Acesso em 15 nov. 2011.

Guia do Hardware. Disponível em: <<http://www.guiadohardware.net/termos/java>>. Acesso em 15 de novembro de 2011.

KATO, Hirokazu, et al. - **ARToolKit 2.33. Documentation.** Seattle, 2000.

KIRNER, Claudio. - TORI, R. **Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade.** In: Claudio Kirner; Romero Tori. (Ed.). *Realidade Virtual: Conceitos, Tecnologia e Tendências.* 1ed. São Paulo, 2004, v. 1, p. 3-20.

KIRNER, Claudio. **Realidade Virtual e Aumentada.** Disponível em: <http://www.ckirner.com/realidadevirtual/?REALIDADE_AUMENTADA>. Acesso em 15 nov. 2011.

KIRNER, Claudio. - **Realidade Virtual: Dispositivos e Aplicações.** Anais da VII Escola Regional de informática da SBC Regional Sul - VII ERI, SBC, Londrina, Chapecó, Novo Hamburgo, maio de 1999, p. 135-158.

LAMB, PHILIP **ARToolKit**. Washington, 2009. Disponível em: <<http://www.hitl.washington.edu/artoolkit/>>. Acesso em 15 nov. 2011.

LIMA, J. P. S. do M. **UM FRAMEWORK DE REALIDADE AUMENTADA PARA O DESENVOLVIMENTO DE APLICAÇÕES PORTÁVEIS PARA A PLATAFORMA POCKET PC**. 2007. 49 F. Trabalho de conclusão de curso (graduação em ciência da computação) – Universidade Federal de Pernambuco, Recife, 2007.

MATOS, Raul César do Carmo. **Projeto de um Protótipo de Aplicação Web com Realidade aumentada**. 2010. 78 f. (Graduação em Informática para Gestão de Negócios) – Faculdade de Tecnologia da Zona Leste, Centro Paula Souza. Disponível em: <<http://fateczl.edu.br/TCC/2010-2/TCC-010.pdf>>. Acesso em 15 nov. 2011.

Nokia Mobile Codes. Disponível em: <<http://mobilecodes.nokia.com/index.htm>>. Acesso em 15 nov. 2011.

NyARToolkit- ARToolKit Class Library for Java/C#/Android. Disponível em: <<http://nyatla.jp/nyartoolkit/wiki/index.php?NyARToolkit%20for%20Java.en>>. Acesso em 15 nov. 2011.

PRESSMAN, R. S. **Engenharia de Software**. 6. ed. São Paulo: McGraw-Hill, 2006. 720 p.

QR CODE, O que é. **Oficina da Net**. Disponível em: <http://www.oficinadanet.com.br/artigo/celulares_e_telefonia/o_que_e_qr_code>. Acesso em 15 nov. 2011.

Realidade Aumentada. Disponível em: <<http://www.realidadeaumentada.com.br>>. Acesso em 15 nov. 2011.

SILVA, Réryka Rubia P.C.L. **Ambiente Colaborativo com Realidade Aumentada**. 2006. 93 f. (Mestrado em Ciência da Computação) – Faculdade de Ciências Exatas e da Natureza, **Universidade Metodista de Piracicaba**. Disponível em: <<http://www.unimep.br/phpg/bibdig/pdfs/2006/CDSMFSPRWSTN.pdf>>. Acesso em 15 nov. 2011.

APÊNDICE A – ENGENHARIA DE SOFTWARE DO SISTEMA

1. Requisitos funcionais

RF #1: Os marcadores deverão ser impressos no formato de um cubo, ou outra forma geométrica que possibilite a impressão de um marcador em cada face e que seja uma peça que ofereça a possibilidade de ser montada a partir de uma matriz de papel.

RF #2: Cada face do cubo deverá possuir um marcador diferente que represente um ambiente diferente.

RF #3: O controle do ambiente virtual deverá ser realizado por um marcador específico para controle.

RF #4: A impressão deverá ser feita em papel fosco, para evitar falhas no reconhecimento do marcador.

RF #5: O computador do usuário deve possuir *webcam* instalada e em perfeito funcionamento para o reconhecimento das imagens.

RF #6: As matrizes de montagem do cubo deverão ser disponibilizadas pela divulgadora por meio de folders com a matriz para recortar e montar e as instruções de montagem, e por meio do site, através do download de um arquivo PDF (*Portable Document Format*) contendo a matriz e as instruções de montagem além do marcador de controle.

RF #7: Durante a simulação o usuário poderá definir o ângulo de visão, ampliar a visão (aplicar ou retirar o *zoom*) e rotacionar a sua visão, por meio de movimentos com o marcador de controle.

RF #8: O sistema deve apresentar a interface e as instruções de uso em português.

2. Requisitos Não funcionais

RNF #1: O sistema deve ser implementado na linguagem FLASH, visando a construção de um sistema flexível quanto a plataforma a ser utilizada. O sistema pode rodar em diferentes sistemas operacionais, por exemplo: Windows ou Linux, e ter versões para dispositivos móveis.

RNF #2: O sistema deve fornecer ao usuário uma interface amigável e intuitiva, que obedeça aos princípios básicos da Realidade Aumentada.

3. Limitações do sistema

LS #1: O sistema requer o sistema Operacional Windows XP ou Linux equivalente, com Adobe Flash Player 9 ou superior instalado e navegadores Microsoft Internet Explorer 8, Mozilla Firefox 3 e Google Chrome 4 ou suas respectivas versões superiores.

LS #2: O sistema requer uma câmera com qualidade VGA (640X480 pixels) ou superior.

LS #3: O sistema requer uma conexão banda larga de no mínimo 256Kbps.

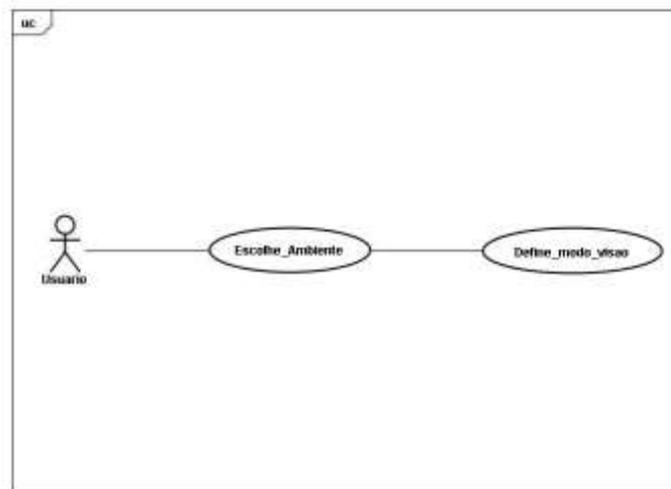
4. Requisitos de desempenho:

RD #1: O sistema deverá ser executado em tempo real.

5. Diagrama de Casos de Uso

Nas práticas de UML, o desenvolvimento dos diagramas de Caso de uso tem por finalidade a modelagem do comportamento do sistema.

No sistema em questão, foi identificado apenas um ator: o Usuário do sistema, que irá interagir com ele.



O usuário escolherá o lado correspondente ao ambiente que deseja exibir, então definirá o modo de visão desejado, aplicando movimentos de rotação.

É composto pelos casos de uso Escolhe_Lado e Define_modos_visao.

Descrição detalhada:

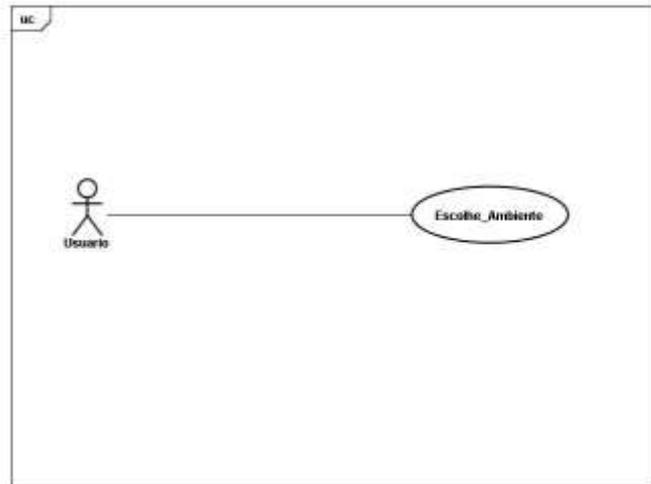
Escolhe_Ambiente

Ator: Usuário

Pré-Condições: A aplicação deve ter sido iniciada

Curso normal: O usuário acessa o site da aplicação e exibe à câmera o lado do cubo correspondente ao ambiente desejado.

Curso de exceção: Se não houver um marcador válido, o ambiente não será exibido.



Define_Modo_Visao

Ator: Usuário

Pré-Condições: A aplicação deve ter sido iniciada e o ambiente deve ter sido escolhido.

Curso normal: Após ter escolhido o ambiente, o usuário retira o cubo do ângulo de visão da câmera e define o modo de visão de acordo com a rotação e a inclinação aplicada ao marcador de controle.

Curso de exceção: Se não houver um marcador válido, a rotação não será aplicada ao ambiente.

