



CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO

EURICO KADOTA
JOSÉ SÉRGIO
RAFAEL SHODI
WESLEI BERNARDES

REALIDADE AUMENTADA

São Paulo
2013

**EURICO KADOTA
JOSÉ SÉRGIO
RAFAEL SHODI
WESLEI BERNARDES**

REALIDADE AUMENTADA

Monografia apresentada junto ao curso de Sistemas de informação da Universidade de Santo Amaro, que tem como requisito parcial à obtenção do título de bacharel.

Orientador: Professor Dr. Aparecido Edilson

São Paulo
2013

EURICO KADOTA
JOSÉ SÉRGIO
RAFAEL SHODI
WESLEI BERNARDES

**REALIDADE AUMENTADA
TRABALHO DE CONCLUSÃO DE CURSO**

Monografia apresentada para obtenção do título de Bacharel em Sistemas de Informação ao Curso de Sistemas de Informação da Universidade de Santo Amaro – UNISA sob orientação da Área de Concentração_____.

Data de Aprovação: ____/____/____

BANCA EXAMINADORA

(nome do orientador e Titulação)

(nome do professor e Titulação)

CONCEITO FINAL: _____

DEDICATÓRIA

*“ Dedicamos este trabalho de conclusão de curso aos nossos pais que sempre nos apoiaram nos momentos mais difíceis, aos nossos professores que nos ensinaram que por mais que achamos que o nosso conhecimento já está bem profundo, estamos enganado pois o conhecimento é algo que está sempre se renovando, e a Deus acima de tudo
Obrigado!”*

AGRADECIMENTOS

Aos Professores Aparecido Edilson e Odair Oliveira de Sá, cujo apoio tem nos permitido realizar nossos ideais de profissão e vida, dedicamos nossa amizade, admiração e respeito.

Agradecemos ainda às pessoas que contribuíram com nosso trabalho de conclusão, e participaram da elaboração deste trabalho trocando informações e experiências.

“A mente que se abre a uma nova ideia jamais volta ao seu tamanho original”

- Albert Einstein

Sumário

Objetivo	5
Resumo	6
Abstract.....	7
Introdução.....	8
Realidade Virtual.....	9
Realidade Aumentada	11
Hiper Realidade.....	13
Técnicas de Interação	15
Técnicas de manipulação.....	17
Hardware e Software	18
Requerimentos para criar realidade aumentada	18
<i>QR Code (Quick Response Code)</i>	19
Análise do Ambiente	21
Metáforas	24
Projeto	26
Biblioteca de realidade aumentada.....	30
Biblioteca de renderização	31
Recursos que podem ser utilizados em engine de renderização	31
Medidas adotadas para desenhar os objetos na tela	31
Medidas a serem tomadas na engine de física para o funcionamento	32
Cadeia de execução das (três) bibliotecas juntas	33
Sistema de reconhecimento	34
Outros tipos de rastreamento	35
Como desenvolver a realidade aumentada	36
Reconhecimento do marcador e animação do objeto	37
Renderização dos gráficos e posições e sistema de coordenada do marcador	39
<i>Software</i> de representação	43
Instalação.....	44
Objetivo da aplicação.....	45
Codificação.....	45
Conclusão	52
Glossário de Siglas e Abreviaturas.....	54
Referências Bibliográficas	55

Índice de Ilustrações

Figura 1 - Demonstração de ambiente de Realidade Virtual	10
Figura 2 - Demonstração de Realidade Aumentada	11
Figura 3 - Demonstração de interação com realidade aumentada	12
Figura 4 - Hiper Realidade - Cena do filme Homem de Ferro	13
Figura 5 - Demonstração de técnicas de interação	16
Figura 6 - Processo de leitura de <i>QR Code</i>	19
Figura 7 - Exemplo de <i>QR Code</i>	21
Figura 8 - Trabalhando com modelagem.....	32
Figura 9 - Cadeia de execução	33
Figura 10 - Rastreamento.....	34
Figura 11 - Identificação do marcador	37
Figura 12 - Correção de distorção de perspectiva.....	39
Figura 13 - Sistema de coordenada	41
Figura 14 - Interface Realidade Aumentada adaptada à Arquitetura do NyARToolkit	43

Objetivo

Neste trabalho de conclusão de curso abordaremos de forma clara e objetiva o funcionamento de uma nova tecnologia que vem ocupando seu espaço aos poucos no mundo tecnológico e de interatividade com as pessoas, conhecida como Realidade Aumentada.

Temos como objetivo apresentar a atuação desta tecnologia que pode combinar objetos criados virtualmente e outras informações com cenários de ambientes reais, e também, ambientes criados virtualmente através de codificação e computação gráfica.

Resumo

Estaremos exemplificando os pontos-chaves de como a Realidade Aumentada e Virtual é trabalhada entre suas diferenças.

Em Realidade Aumentada e Virtual iremos abordar a forma de como é realizado o rastreamento, o desenvolvimento do ambiente tridimensional, seja ele feito virtualmente ou mesclado com a realidade aumentada.

Também iremos abordar de forma breve e resumida o funcionamento do novo modo de código de barras, mais conhecido como QR Code, como é feita a leitura do QR Code e como o sistema se comporta no processamento e busca da animação que irá ser gerado na tela.

Entramos também na questão do hardware, seus requisitos para que possa ser desenvolvido e demonstrado ao usuário final o tridimensionamento da imagem, assim como as bibliotecas necessárias para o desenvolvimento de uma aplicação simples e funcional.

Por fim criamos um passo a passo de como desenvolver uma aplicação para realidade aumentada, criando detalhadamente quais bibliotecas são necessárias e onde as mesmas podem ser baixadas, classes a serem utilizadas e qual será o objetivo da aplicação.

Abstract

We will be illustrating the key points of how Augmented Reality and Virtual is crafted from their differences.

Virtual and Augmented Reality will address how to trace how it's performed, the development of three-dimensional environment, whether it be done virtually or merged with augmented reality.

Also we discuss briefly summarized and the operation of the new mode of barcode , known as QR Code, how it is done reading the QR Code and how the system behaves in processing and searching of the animation that will be generated on the screen.

Also entered the hardware issue, its requirements for development and that can be shown to the end user tridimension the image, as well as the libraries needed to develop a simple application and functional.

Finally we have created a step by step how to develop an application for augmented reality , creating detail what libraries are needed and where they can be downloaded, classes to be used and what is the purpose of the application.

Introdução

Antes do surgimento das máquinas há longos anos atrás, a interação das pessoas em seu dia-a-dia era somente em ambientes naturais, reais. Como não se era comum na época, em alguns momentos as pessoas entravam em algum tipo de interação com as máquinas, seja ela qual for o tipo de ação.

Com o passar dos anos e com a evolução dos computadores, a interação das pessoas com as máquinas começou a se intensificar ano pós ano devido o surgimento das necessidades que as pessoas encontravam em seus dia-a-dia, seja ela em suas tarefas de trabalho, lazer e/ou profissional.

Mesmo com a atual tecnologia disponível para as pessoas comuns (domesticas e profissionais sem foco na tecnologia) incluindo *web cams*, impressoras, *tablets*, *mouses* e teclados sem fio a crescente necessidade ainda surge nas pessoas, na facilidade de se interagir entre a interface Homem x Máquina.

A nova tendência do mercado tecnológico está sendo focado no tema que iremos abordar a seguir, chamado de Realidade Aumentada, surge como um novo meio de interação entre pessoas e máquinas, rompendo barreiras que hoje é mais conhecido como Telas de Computadores, tornando-as apresentações mais reais tridimensionalmente mais próximo da realidade entre os usuários.

A realidade aumentada surgiu em meados da década de 60 pelo desenvolvedor Ivan Sutherland, devido as grandes dificuldades dos usuários a se adaptarem a realidade virtual, onde o mesmo necessitava de equipamentos como capacete, luva, óculos estereoscópicos, *mouses*, 3D, etc.

Porém, os estudos e os pensamentos para tornar a tecnologia mais acessível só ganharam maior atenção em meados da década de 90 quando a tecnologia era mais avançada comparada ao ano de criação da tecnologia, o que proporcionou estudos mais profundos e avançados com meios tecnológicos mais práticos e bem fundamentados.

Realidade Virtual

A Realidade Virtual possibilita que o usuário interaja com um ambiente virtual gerado por um computador, em tempo real.

Para a elaboração de interfaces de ambientes virtuais são utilizadas bibliotecas de Realidade Aumentado como VRML, NyARToolkit, OpenGL 2D e 3D, entre outras, *softwares* de renderização como Blender e linguagem de programação como Programming.

Estes ambientes permitem aos usuários de forma intuitiva visualizar, manipular objetos, e explorar ambientes virtuais, e que o próprio ambiente reaja conforme a ações dos usuários trazendo assim uma maior imersão no ambiente construído. A interação do usuário com o ambiente virtual pode-se utilizar de equipamentos como luvas para realizar gestos, capacetes para visualização e audição do ambiente, microfones para comandos de voz, ou ainda o próprio corpo para ter seus movimentos captados e reproduzidos no ambiente.

Outros dispositivos de interação mas que não possibilitam uma imersão tão grande são os já tão conhecidos como monitor, teclado e *mouse*.

Existe também uma forma de navegação que fornece apenas a possibilidade de navegar em um ambiente, porém sem efetuar alterações ou esperar que o mesmo se modifique, esta forma é chamada de Tour Virtual ou simplesmente Passeio Exploratório.

Para que o usuário possa ter uma experiência se faz necessário que exista a disponibilidade de um bom conjunto de *hardware* para evitar que o tempo de resposta aos comandos do usuário ao ambiente virtual não ocorra com atraso. Para que ainda seja considerada uma experiência em tempo real, o atraso não deve superar a margem de 100ms. É desejado que o *hardware* disponibilizado tenha ótima capacidade de processamento gráfico para que este seja capaz de trabalhar com uma taxa de renderização por volta de 20fps (Quadros por segundos), assim reproduzindo o ambiente tridimensional, que é calculado em tempo real, sem oferecer atrasos, ou que estes sejam quase imperceptíveis não atrapalhando a navegação e interação.

A experiência do usuário em um ambiente virtual pode não ser imediatamente funcional ao usuário, e este pode necessitar de um tempo para se adaptar, ou até mesmo pode ser necessário que o usuário seja treinado para que assim ocorra um melhor aproveitamento de todo conteúdo criado.

Figura 1 - Demonstração de ambiente de Realidade Virtual



Fonte: <http://www1.folha.uol.com.br/folha/ciencia/ult306u624280.shtml>

Realidade Aumentada

A Realidade Aumentada possibilita a inclusão de informações, gráficos, objetos, imagens, efeitos de som, todos virtuais a um ambiente real de forma a completar aquele ambiente, possibilitando que o usuário possa obter e interagir com as informações geradas apresentando uma forma adicional de se enxergar o ambiente podendo facilitar suas atividades ou apenas enriquecendo a experiência naquele ambiente.

Figura 2 - Demonstração de Realidade Aumentada



Fonte: <http://www.itouchbr.com.br/2013/09/04/especial-realidade-aumentada/>

Figura 3 - Demonstração de interação com realidade aumentada



Fonte: <http://www.digitalbeat.pt/sobre-a-realidade-aumentada/>

Hiper Realidade

Uma experiência de Hiper Realidade visa potencializar a interação entre o usuário, ambiente real e as informações virtuais geradas. O ambiente proporciona a interação do usuário com pessoas, animais, construções, objetos, para assim suprir qualquer necessidade de informação, como uma pessoa que passeando por uma cidade e utilizando-se de um dispositivo que lhe insira nesse ambiente, possa obter informações dos estabelecimentos e ruas, visualizar objetos virtualmente dispostos em determinados pontos da cidade, interagir com animais virtualmente dispostos, tudo em tempo real com integração de informações obtidas pela internet, ampliando a experiência do usuário e lhe possibilitando maiores fontes de informações e sensações com todo material gerado.

Figura 4 - Hiper Realidade - Cena do filme Homem de Ferro



Fonte: <http://antenadius.wordpress.com/2012/09/14/tecnologia-ubiqua/>

Rastreamento

A necessidade de rastreamento para ambientes de realidade virtual e aumentada se faz para a captura dos movimentos do usuário, assim como a identificação e localização de seus membros, como pernas, mãos, cabeça, etc. Com os membros identificados e os movimentos capturados, o usuário pode exercer comandos como segurar, apertar, empurrar, mover, entre outros, apenas a partir de seus gestos.

Os dispositivos disponibilizados para o rastreamento e interação com ambientes virtuais, em geral, são dispositivos de custo elevado, o caso dos de rastreamento, são dispositivos magnéticos, mecânicos, etc. O custo vem sendo diminuído com o avanço dos processadores e com a facilidade de utilização de *webcams*.

A vantagem de se utilizar câmeras para o rastreamento é que estas são de baixo custo, não necessitam de interação física com o usuário, e apenas necessitam conhecimento das variáveis e parâmetros de rastreamento.

Técnicas de Interação

É nítida a evolução das técnicas de interação do ser humano ao longo de sua história. Desde o início com pinturas nas cavernas, passando pela criação do jornal, rádio, televisão e enfim o computador, o ser humano sempre buscou novas formas de interagir com as informações de seu meio ambiente, enriquecendo cada vez mais sua busca e disseminação de informações.

Com o surgimento do computador a forma de interação ampliou seus horizontes, pois agora é possível realizar interações com diversas aplicações e dispositivos.

Com a diversidade de aplicações, é de grande importância o desenvolvimento de interfaces que facilitem a interação do usuário com a aplicação. Aplicações mal projetadas ou confusas podem ocasionar desinteresse por parte dos usuários. Essa busca pelo bom desenvolvimento de uma interface intuitiva que facilite a comunicação com o usuário toma boa parte do processo de planejamento e desenvolvimento. De qualquer forma, uma aplicação com uma boa interatividade tem seus pontos positivos como a aprendizagem de manuseio do usuário em um tempo menor e um maior aproveitamento dos recursos da aplicação.

Em aplicações de ambientes 2D, o usuário interage com o software utilizando teclado e mouse, realiza cálculos, cria textos, imagens, vídeos, músicas, etc. Além dos métodos de interação com ambientes 2D, a interação com ambientes 3D demandam técnicas que trabalham com espaço.

A interação maior ocorrer em ambientes de Realidade Aumentada, onde o usuário tem maior participação não se limitando apenas ao hardware do computador. Este, o usuário, pode interagir através da captura de seus próprios movimentos enviando comandos ao computador o qual identificará a ação ou o objeto exibido e irá projetar uma nova informação no ambiente.

As técnicas de interação possuem três objetivos como principais, o Desempenho, que trata como esta a qualidade da interação entre o computador e o usuário, a Usabilidade, que trata como esta sendo a facilidade do uso da aplicação e

a aprendizagem deste de uso por parte do usuário, e Utilidade, que trata de como o usuário esta atingindo seus objetivos em relação a aplicação.

Figura 5 - Demonstração de técnicas de interação



Fonte: <http://www.xbox.com/pt-PT/Kinect>

Técnicas de manipulação

- Manipulação 3D: A técnica pode ser utilizada de maneira intuitiva ou não, a interação com o usuário deve ser realista, fazendo com que o usuário realmente sinta que está manipulando um objeto real.

Na implementação desta técnica, deve ser verificado como indicar um evento de manipulação. É importante que o sistema retorne ao usuário alguma informação para que este saiba que o objeto está disponível ou que foi selecionado corretamente. E após a manipulação do objeto, o ambiente retorne uma informação atualizando a situação do ambiente após a interação com o objeto.

- Controle do sistema: Para o controle do sistema, as técnicas de interação que são utilizadas são comandos de voz, menus gráficos, comandos por gestos, entre outros. Menus em 2D também podem ser aplicados em ambientes 3D, e manipulados através dos comandos citados.

As implementações do controle de sistema visa facilitar a interação do usuário com o ambiente, tornando que as ações dos comandos sejam feitas sem que o usuário tire o foco do objetivo principal.

- Entrada simbólica: A interação através de entrada simbólica pode ser baseada em gestos, fala, teclado ou teclado virtual, canetas, luvas, etc.

- Técnicas específicas para Realidade Aumentada: A técnica possibilita que o usuário possa interagir tanto com objetos virtuais quanto com reais enquanto utiliza a aplicação.

Hardware e Software

O avanço das aplicações na área de Realidade Aumentada se deve muito a diminuição dos custos de *hardware* e a disponibilização de *softwares* livres para o desenvolvimento, onde facilita que desenvolvedores trabalhem em novas aplicações aperfeiçoando técnicas e *softwares* relacionados.

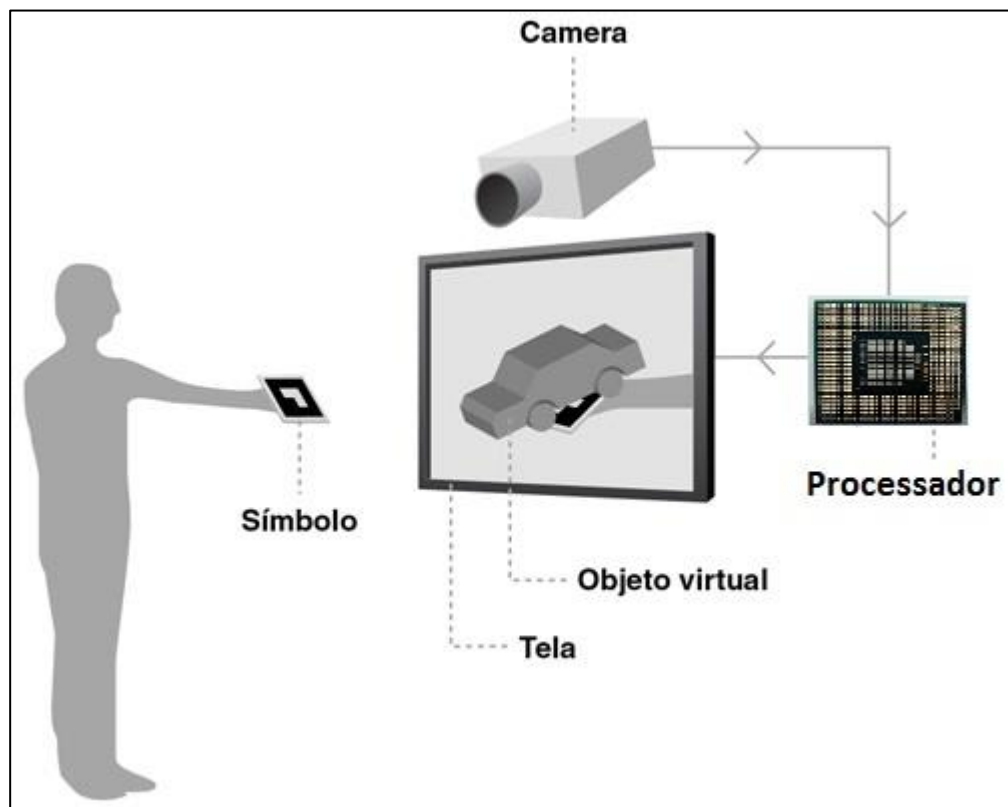
O *hardware* tem uma atuação de grande importância, pois será este que irá suprir as necessidades da aplicação, a qual irá exigir um bom desempenho de processamento. O conjunto de *hardware* é consistido de diversos periféricos como sensores, placa de vídeo, processador, placa de rede, entre outros.

Requerimentos para criar realidade aumentada

- Câmera
- Tela
- Processador

Uma das grandes vantagens da tecnologia de Realidade Aumentada é o pouco investimento em dispositivos de *hardware*. Basicamente para podermos iniciar um projeto de RA, precisamos apenas de uma câmera para captura e reconhecimento do ambiente, um processador para o processamento dos dados de entrada e uma tela para encher os olhos do usuário com apresentações gráficas misturando o ambiente virtual com o ambiente real. O usuário ainda pode interagir com os objetos virtuais como se estivesse no mundo real. Veja a ilustração da figura abaixo.

Figura 6 - Processo de leitura de QR Code



Fonte: <http://library.epfl.ch/en/tools/?pg=qrcode>

QR Code (Quick Response Code)

QR Code é considerado uma evolução do código de barras já conhecido, criado em 1994 tendo sua patente registrada pela Denso Wave, empresa integrante do grupo Toyota com a finalidade de identificar as peças de seus veículos. O QR Code pode ser lido através de uma câmera e convertido em texto, e este pode trazer um e-mail, um número de telefone, uma url entre outros, e tem a capacidade de armazenar cerca de 4.2 mil caracteres alfanuméricos.

Atualmente, este código é encontrado facilmente em propagandas e cartões de visita, que quando escaneado fornece mais informações sobre determinado produto ou serviço, informações que podem ser imagens adicionais,

uma melhor descrição, inserção de telefone na agenda do seu celular, e projeções 3D para realidade aumentada.

Os primeiros anúncios com fins publicitários realizado no Brasil foram feitos pela rede de lojas Fast Shop em 2007.

A estrutura de um *QR Code* contém elementos que identificam a versão, o formato, dados e chaves de correção, padrões, posição, alinhamento e o tempo. Existe uma margem para que o leitor saiba identificar os limites do código, os quadrados maiores nos cantos, que podem aparecer uma, duas ou três vezes no mesmo QR Code, servem como orientadores, identificando seus cantos, ao lado destes quadrados maiores, há uma área onde identifica a versão do código, que pode variar de 21x21 à 177x177 módulos. Na parte interna dos quadrados maiores, há um pontilhado que ajuda a identificar linhas e colunas, e o quadrado menor, encontrado no canto onde não constam os maiores de orientação, é uma referência de alinhamento, o restante do código é o conteúdo em si.

O Código QR armazena informações que torna possível sua leitura mesmo que este esteja danificado, possuindo alguns níveis referentes ao tratamento e correção.

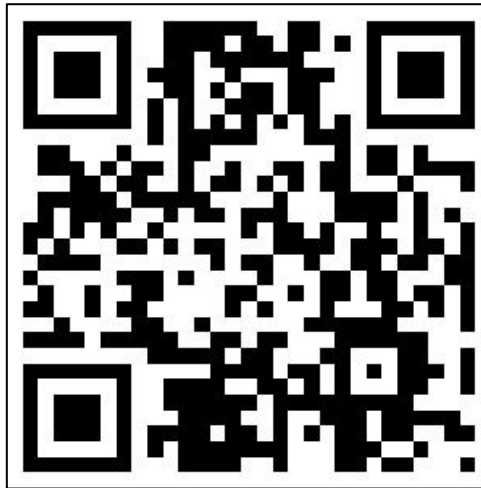
Nível L (Low / Baixo) 7% de capacidade de correção;

Nível M (Medim / Medio) 15% de capacidade de correção.;

Nível Q (Quartile / Quartil) 25% de capacidade de correção.;

Nível H (High / Alto) 30% de capacidade de correção.

Figura 7 - Exemplo de QR Code



Fonte: <http://g1.globo.com/tecnologia/>

Análise do Ambiente

Realidade Virtual entende-se que a tecnologia avançada da interface, permite o usuário fazer imersão, navegar e interagir em ambiente sintético tridimensional gerado pelo computador, usando canal multissensoriais. A vantagem da RV de outra forma de interagir humano-computador, é o ambiente que é visualizado de qualquer ponto de vista e a medida que vem sendo feita alteração em tempo real. Suportar esse tipo para interagir, o usuário usa dispositivos não convencionais, típicos da área de RV, como capacete para visualizar e controlar, luvas, *joystick* ótico, etc.

O dispositivo dá ao usuário a impressão de que o aplicativo funciona no ambiente tridimensional real, permitindo a exploração do ambiente e manipulação natural dos objetos. Ambiente envolvido em determinado espaço e situação delimitada, inserindo todos os componentes neles inseridos, como o conjunto de objetos e de condições passíveis de serem percebidos e com qual é possível interagir. Ambiente virtual é ambiente interativo, gerado por computador e disponibilizado por um sistema de realidade virtual.

De acordo com Martins (2000) e Vince (2004), um ambiente virtual típico deve incluir características que tornem: Sintético: Significa que o ambiente é gerado em tempo real pelo sistema computacional. Não é, por exemplo, pré-gravado, como acontece com sistema de multimídia.

Tridimensional: Significa que o ambiente que cerca o usuário é representado em três dimensões em 3D, que existem recursos que ajuda a ter ideia que o ambiente possui profundidade e que o usuário pode se movimentar através dele.

Multissensorial: Significa que mais de uma modalidade sensorial é usada pra representar ambiente, com sentido visual, sonoro, espacial, reação do usuário com o ambiente, etc.

Imersivo: Entende que aqui, mais do que olhar e ouvir um *display* vindo do monitor, o *display* necessita criar impressão de que se está dentro do ambiente produzido computacionalmente. Normalmente, um sistema imersivo é obtido com uso de capacetes para visualizar, mas outros sentidos, como som e controles reativos, são também importantes.

Interativo: Refere à capacidade do computador detectar as entradas do usuário e modificar na hora o mundo virtual com ações realizadas sobre ele.

Realístico: Envolve a precisão com que o ambiente virtual reproduz objetos reais, as interações com os usuários e o próprio modelo do ambiente.

Com presença: Caracteriza como sendo um sentido subjetivo, responsável por oferecer ao usuário a impressão de que ele está fisicamente dentro do ambiente virtual.

Para qualquer sistema de *software*, a construção dos ambientes virtuais requer processo de desenvolvimento adquirindo às suas peculiaridades. O processo segue uma abordagem de desenvolvimento interativo dos sistemas de *software*. Abordagem combina características de desenvolvimento, buscando a geração da versão que vão sendo sucessivamente refinados, até chegar a uma implementação que atende os requisitos inicialmente identificados. As etapas e fases que inclui o processo para desenvolvimento, compreende desde a

definição dos requisitos até a implementação e avaliação do sistema. O desenvolvimento dos ambientes virtuais e de realidade aumentada requer esforço conjugado envolvendo temas relacionados com a modelagem gráfica 3D e interface humano-computador (IHC), a sua construção é ligada ao realismo visual e a interação pelos outros sentidos humanos distintos.

O processo explora o máximo os recursos de RV e IHC, modelagem gráfica e visualização de informação para validar este processo de desenvolvimento.

Objetivos principais da abordagem são:

- Liberar a participação dos usuários, para que eles possam experimentar o *software* que está sendo desenvolvido e opinar sobre possíveis melhorias.
- Criar um sistema que possa ser utilizável e de fácil manutenção, pelo período de tempo adequado e o custos aceitáveis.

Realização dos objetivos buscados requer, principalmente:

- Os projetistas saibam quem serão os usuários, identificando em termos de atitudes, comportamentos e necessidades.
 - A equipe de desenvolvimento atue em conjunto com os usuários.
 - Protótipos sejam criados rapidamente, avaliados pelos usuários e consecutivamente corrigidos, refinados e melhorados.
- Processo de Desenvolvimento de Ambientes Virtuais Exigências. Processo de desenvolvimento de software para aplicação de realidade virtual baseia em duas exigências essenciais.

Ambiente virtual e sua interface devem ser adaptados para a tarefa, essa exigência está inserida no reconhecimento do fato de que a imersão, no ambiente tridimensional gerado pelo computador, mostra oportunidades que não são encontradas na computação gráfica convencional. Por tanto, é necessário que se analisem questões de todos os projetos, o projeto da interface e questões importantes de fatores humanos.

- Limites rigorosos do desempenho devem ser cumpridos para que os benefícios de RV sejam alcançados. Isso se refere ao fato de ser utilizável, admitindo a taxa adequada de quadros / segundo suficiente para não frustrar o usuário.

Essas duas exigências estão, em conflito, mas o desenvolvimento bem sucedido de uma aplicação de RV deve atender, simultaneamente a ambos os requisitos. O desenvolvimento do aplicativo de RV deve ter tanto uma abordagem.

Metáforas

A maneira de como um aplicativo será implementado dependerá muito de cada aplicação. Algumas questões podem ser discutidas levando em conta as metáforas em RV que são adotadas. Metáfora significa como o usuário se relaciona com o ambiente virtual. A ideia é liberar ao usuário pensar em termos de interação com objetos que são diretamente relacionados a tarefas disponíveis, em vez de pensar em termos de interagir com o computador.

Realidade Virtual, o usuário está imerso no ambiente e as metáforas convencionais, como interface gráfica do usuário, estão fora do ambiente, apropriadas para tais sistemas. Construir boa metáfora para o ambiente virtual, numa aplicação, a tarefa crítica, visto que é necessário saber como está metáfora determinará a aparência e o comportamento do ambiente, como o usuário interagirá com tal ambiente. Uma metáfora permitirá ao usuário interagir eficaz e confortavelmente com ambiente virtual para executar a tarefa da aplicação. Evidente que uma única metáfora não cobrirá todas as aplicações de RV ou todos os domínios, sendo necessário que estas metáforas sejam construídas com base em cada caso. A metáfora também, não deve derivar do mundo real, mas de uma linguagem convencional do domínio da aplicação para que o ambiente seja construído.

Pode definir vários níveis de uma metáfora em um ambiente virtual:

- Metáfora para todo o ambiente: Determina a aparência de todo ambiente, incluindo os tipos de objetos que aparece no ambiente e tipos de comportamentos.

- Metáfora para representação de informação: Como a informação deve ser apresentada ao usuário.

- Metáfora de interação: Como o usuário interage com ambiente e com os objetos no ambiente.

O nível de metáfora pode conter várias metáforas. A metáfora de representação de informação pode adicionar textos que aparece no ambiente (talvez em uma janela de informação ou texto 3D), como a informação de interação pode incluir a manipulação direta de objetos, tais como seleção e botões do menu através de controle.

Experiências com RV e computação gráfica, mostra que pode ser interessante mesclar metáforas de duas áreas, facilitando a usabilidade do ambiente virtual.

Engenharia de requisitos tem como objetivo identificar e descrever as necessidades de informação do ambiente virtual pretendido.

- Elicitação dos requisitos: As ações que visam capturar e registrar as informações necessárias para o entendimento correto e completo das necessidades e expectativas para o entendimento correto e completo da necessidade e expectativa dos usuários.

É importante considerar três elementos essenciais:

Potenciais usuários do sistema, identificar aspectos como a características específicas dos usuários, por exemplo, se já possui experiência com o uso de ambiente virtual, a aplicação será mono ou multiusuário, se os usuários acessarão o ambiente virtual remotamente, se será um ambiente colaborativo.

Tarefas a serem executadas no ambiente virtual, como essas tarefas podem ser realizadas. Os usuários podem caminhar pelo ambiente virtual, pegar e arrastar objetos, ouvir determinados sons. Aplicação que pode ser definido sob

diferentes aspectos, ambiente físico, ambiente de trabalho, ambiente social... Para ambiente virtual, a definição do ambiente físico em que o sistema funcionará deve merecer atenção especial. Ele pode ser definido com maior ou menor grau de realismo, dependendo da situação dos usuários e da restrição.

Especificação dos requisitos: Envolve principalmente a modelagem e descrição detalhada dos requisitos. Definição dos requisitos gerais do ambiente virtual é geralmente representado através de modelos conceituais.

Definição dos requisitos, ligados à característica intrínsecas da RV. É importante analisar três possíveis rotas: quando a tarefa, ser executada corresponde a do mundo real, quando a tarefa for executada da maneira diferente do mundo real, e quando a tarefa não corresponde a tarefa do mundo real.

Definição dos requisitos não funcionais são relacionados a um aspecto de *software*, de *hardware* e de fator externo, que determina condição ou restrição a um comportamento do ambiente virtual considerado. Levam em conta os fatores de qualidade exigidos pela aplicação, como usabilidade, portabilidade, confiabilidade... Quando as exigências das aplicações de realidade virtual, como interação, sentido de presença, navegação.

Projeto

Tem por objetivo, especificar as tecnologias de entrada, de saída e computacionais a ser empregada no desenvolvimento do ambiente virtual. [Martins, 2000; Stuart, 1996].

Tecnologias de entrada: Esta tecnologia, cruciais para sistemas de realidade virtual e aumentada, pode adicionar: capacetes, luvas, câmeras de vídeo, rastreadores de posicionamento.

Tecnologias de saída: Objetivo é apresentar informação ao usuário de uma forma que ela possa ser percebida pela tarefa, dando as condições descritas na

definição dos requisitos. Tecnologia de saída pode ser dispositivos visuais, fones de ouvidos, dispositivos de tato.

Tecnologias de *hardware*: São definidos quais equipamentos são empregados, incluindo computadores e dispositivos de realidade virtual.

Tecnologias de *software*: É *software* que atende os requisitos de realidade virtual deve ser: de navegação, interativo, autônomo, que permita o uso de *scripting* e que permita a interação com multimídia.

Projeto de objetos, comportamentos e interações: Serão definidos os objetos, que tem algumas questões como: qual sua geometria, cor, escala, tamanho e textura, os comportamentos que um objeto possa demonstrar, e a interação necessária para o ambiente virtual, como a movimentação, manipulação, seleção, escalonamento, modificação de objetos e de propriedades do mundo.

Implementação

Objetivo construir um ambiente virtual, compreendendo as atividades descritas.

Obtenção e preparação de imagens: As imagens necessárias podem ser obtidas através de um *scanner*, câmera de vídeo ou *software* de desenho.

A ajuda de um editor apropriado, as imagens devem ser preparadas e convertidas para o formato de arquivo compatível com o *software* de realidade virtual.

Construção dos objetos 3D: Objetos geométricos são criados usando os *softwares* de modelagem 3D e ferramentas de criação de mundo virtuais.

Execução do ambiente virtual: Consiste em criar, propriamente, o ambiente virtual. É nesta fase que os objetos 3D são incorporados ao ambiente, assim como os pontos de vista, a luz iluminará o ambiente, a animação, os sensores.

Avaliação

Principais atividades na avaliação são descritas.

Avaliação do desempenho do sistema: Alguns quesitos para avaliar o desempenho do sistema são:

Latência total é o acúmulo de atrasos introduzidos por cada componente do sistema, necessário para se gerar resposta do sistema: taxa de dados e atualização de rastreadores de posicionamento e tempo gasto para executar a simulação.

Taxa de atualização do *display* é a frequência com que sucessivas imagens visuais aparecem no *display* (embora esta taxa contribua pra latência total, deve ser analisada individualmente). Robustez é altamente específica da aplicação, que suas métricas devem ser aplicadas pra cada aplicação.

Tolerância a falhas que não é uma característica específica de sistemas de RV, mas é necessária para um emprego real de aplicação, como tele presença.

Avaliação da usabilidade: Em outras IHCs, avalia a usabilidade, incluindo taxa de erro, carga de trabalho, realização da tarefa e aprendizado, pode contribuir para melhor o projeto. As abordagens mais conhecidas, avaliação empírica é mais usada pra avaliar a usabilidade de ambiente virtual. Pode utilizar técnicas como análise de videotape, entrevistas, questionários e monitoramento psicológico. Através de técnicas, é possível medir as taxas de erros, carga de trabalho, aprendizado, tempo e taxa de realização de tarefas.

Avaliação da segurança: Segurança de *software* é uma atividade que garante a qualidade do *software*. Estas atividades tentam encontrar e avaliar casualidade que possa agir negativamente sobre o *software* e provocar uma falha em todo sistema. As casualidades puderam ser identificadas prontamente no processo de engenharia de *software*, é possível especificar características de projeto

de *software* para excluir ou tratar essa casualidade. A modelagem e análise podem ser usadas como parte da segurança de *software*. Primeiro estas casualidades são identificadas.

Técnica de análise é utilizada para determinar a gravidade e a probabilidade de ocorrência. Finalmente a especificação pode conter uma lista de eventos indesejáveis que devem ser gerenciados (dar respostas desejadas do sistema ao evento).

Avaliação do valor da tarefa e aplicação: Avalia o valor do sistema de RV para a aplicação, o qual foi desenvolvido é necessário primeiramente, que defina um critério em que o valor será considerado. Desde que a análise da tarefa já tenha sido feita durante o processo de especificação de requisitos, uma abordagem é avaliar o sistema através de análise do sucesso pra cada sub-tarefa. Uma falha desta abordagem para avaliar o valor do sistema para aplicação, alguns casos erroneamente, que os usuários realizarão tarefas da mesma maneira que antes, apenas um pouco melhor devido o sistema de RV. Esta abordagem erra quando os usuários podem realizar tarefas com este sistema que não poderiam fazer antes ou da maneira indiscutivelmente diferente da anterior.

Exemplos de ambientes virtuais. A série de ambientes e aplicações de ambientes virtuais foram desenvolvidos, valendo o processo de desenvolvimento proposto nas seções anteriores. Alguns desses ambientes virtuais são destacados.

Museu virtual: É um ambiente virtual que representa um Museu histórico, de visitação pública, representando uma réplica das salas e objetos existentes no museu.

Ambiente de visualização de informação: É um ambiente que combina recursos de RV e visualização de informação, servindo como uma ferramenta de visualização de dados.

Ambiente de comércio eletrônico: Um ambiente virtual que segue uma metáfora de um *Shopping Center*, para vendas através da internet.

Ambiente colaborativo para a educação: É um ambiente multiusuário, baseado na teoria pedagógica do construcionismo, funcionando na internet, para suporte a diferentes aplicações educacionais.

Máquina de medir por coordenadas: Ambiente virtual simula a máquina, sendo de grande utilidade para o ensino da engenharia na área de engenharia.

Ambiente de simulação de sistema de tempo real: Um ambiente virtual para simulação de uma aplicação na área de manufatura.

Visualização de dados: É um ambiente virtual, com realidade aumentada, que suporta a visualização de dados em um sistema de pedágio.

Biblioteca de realidade aumentada

NyARtoolkit é uma biblioteca de código *open source*, desenvolvida para facilitar o desenvolvimento de Realidade Aumentada. Essa biblioteca é responsável por realizar cálculos de posicionamento e projeção dos objetos virtuais no ambiente real, com ela podemos empregar métodos de visão computacional e processamento de imagens capturadas por uma câmera.

Um dos pontos cruciais da NyARToolkit é calcular precisamente a ponto de observação do usuário o ambiente em tempo real. Essa biblioteca é capaz de capturar o ambiente rapidamente e processar as informações de forma muito ágil e a partir daí possibilita o programador a inserir objetos virtuais sobre os objetos do mundo real.

Biblioteca de renderização

OpenGL é uma biblioteca de renderização de gráficos. Segundo o livro Fundamentos e Tecnologia de Realidade Virtual e Aumentada,

A renderização de gráficos de Realidade Aumentada era feita diretamente no *hardware* o que exigia muito do programador para atingir um determinado objetivo com RA. (TORI, KIRNER, SISCOUTO, 1980).

Sabendo da dificuldade e da baixa produtividade nesse ramo da computação, tinha a necessidade do desenvolvimento de alguma API que atendesse esse ramo tão necessitado. Por volta dos anos 90 deu-se inicio a evolução da OpenGL uma biblioteca que carrega um conjunto com varias funções que facilita a programação de RA. Nessa ferramenta existem praticamente todos os recursos de *hardware* e vídeo necessários para a renderização dos objetos virtuais.

Recursos que podem ser utilizados em engine de renderização

- Sombras é o recurso mais importante, porque no objeto virtual ela não aparece enquanto no Objeto real ela aparece claramente.
- Oclusão faz com que o objeto virtual cubra o objeto real.

Medidas adotadas para desenhar os objetos na tela

- Ajuste da câmera para que ela fique na mesma posição que ela está na biblioteca de Realidade Aumentada.
- Matriz de projeção da câmera real ou tranco de visão.

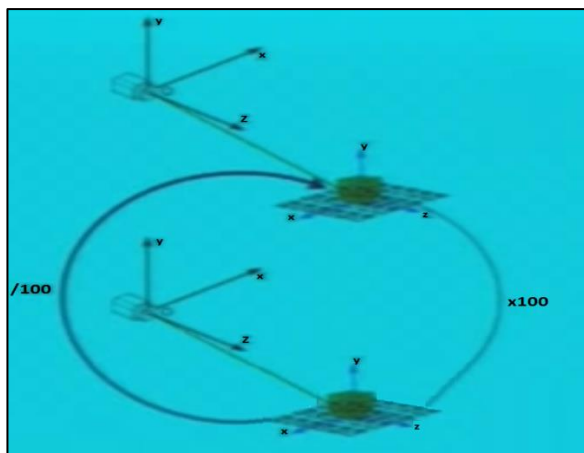
- Engine de física onde podemos adicionar detecção de colisão e dinâmicas de corpos rígidos aos objetos. Essa biblioteca é utilizada para detectar colisões entre os objetos e a partir desse efeito abstraído podemos gerar um movimento qualquer para os objetos que colidiram.

Medidas a serem tomadas na engine de física para o funcionamento

- Sistema de coordenada da câmera terá que transformar para o sistema de coordenada do marcador onde a origem fica no centro do marcador e os eixos alinhados a ele essa transformação precisa de feita de ida e volta.

- Biblioteca de realidade aumentada te dá as medidas em metros e como os objetos são normalmente trabalhados em tamanhos pequenos os valores terão que ser pequenos, só que as engine de física não trabalham bem com valores menores que 1 e ainda teremos que transformar esses valores de metros para centímetros, ou seja multiplicar por 100 e na hora de voltar para engine de renderização dividir por 100 conforme ilustração abaixo.

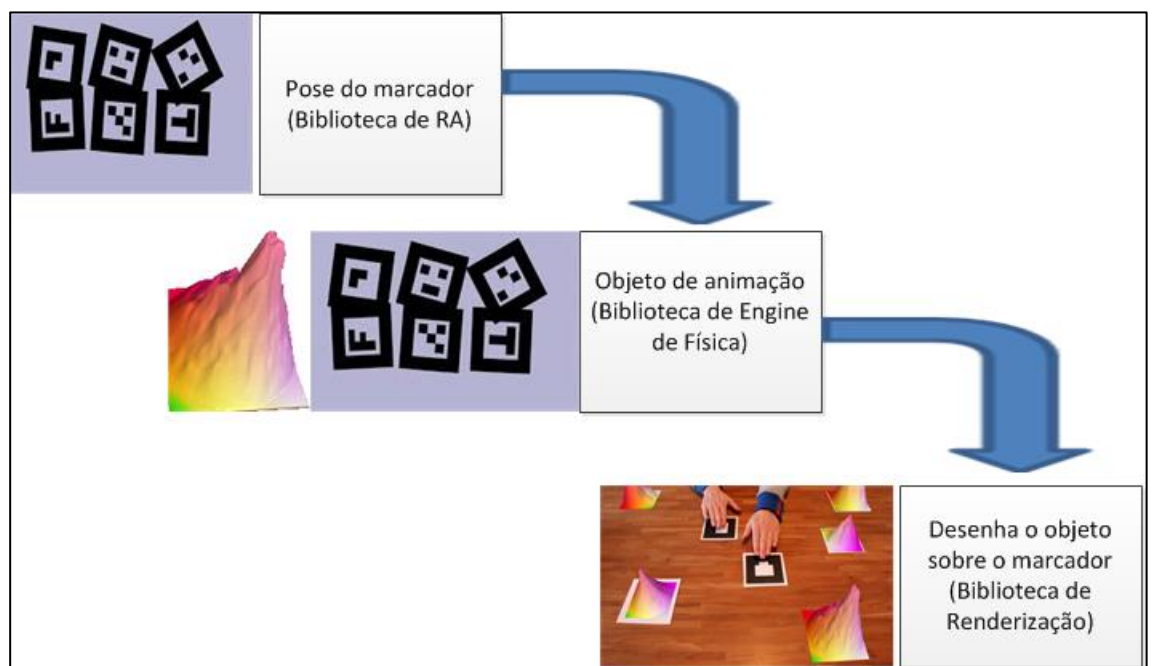
Figura 8 - Trabalhando com modelagem



Cadeia de execução das (três) bibliotecas juntas

- Primeira faz a estimação da pose com a biblioteca de realidade aumentada que nos dar a pose do marcador.
- Segunda faz a simulação do movimento usando a *engine* de física que ela vai nos dar o movimento desse objeto.
- Terceira a partir dessas duas informações nos desenhamos o objeto se movimentando sobre o marcador com a renderização. Veja na ilustração a seguir..

Figura 9 - Cadeia de execução



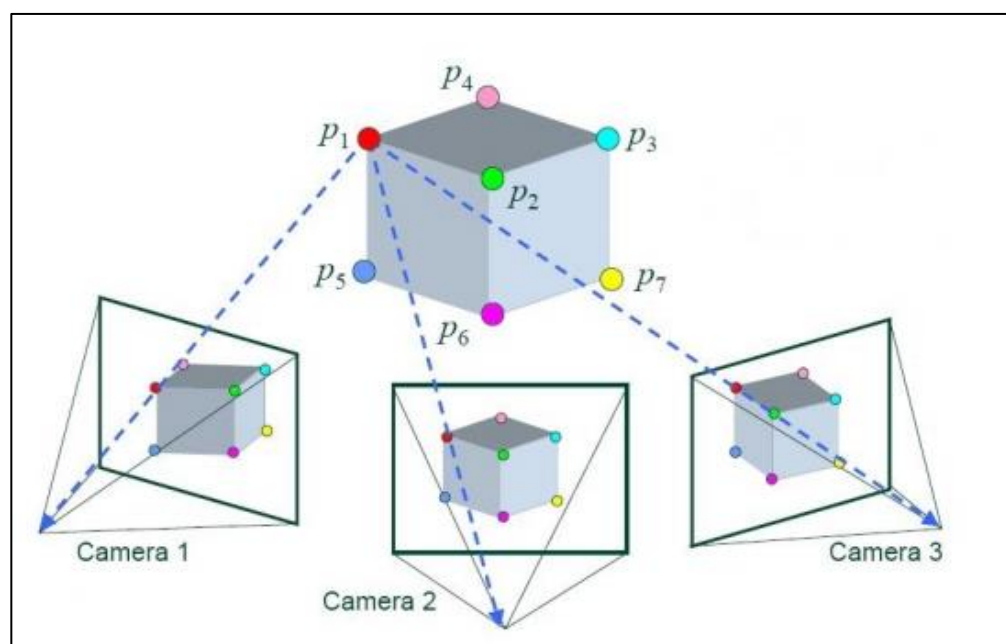
Fonte: Próprio autor

Sistema de reconhecimento

O rastreamento óptico é utilizado para sistemas especializados em Realidade Aumentada e Virtual. Com a leitura óptica é possível realizar o rastreamento do objeto que quer se rastrear, sendo assim, construindo imagens 3D e tendo uma interação bem satisfatória com o usuário. A facilidade disto é a não necessidade de contatos mecânicos para o rastreamento do marcador, porém, para tal rastreamento é necessário ter um conhecimento vasto em programação para entender a forma de como a aplicação irá trabalhar com parâmetros e variáveis que estão engessadas dentro da programação do *software*.

Uma boa parte das aplicações que contém Realidade Aumentada utiliza-se de símbolos de marcação artificial. Esses símbolos servem para ajudar o *software* a indicar onde os objetos virtuais (3D) devem ser inseridos em cenas utilizadas para apresentação, sem essas marcações fica praticamente impossível do *software* reconhecer o ponto e apresentar os objetos 3D no local indicado pelo desenvolvedor.

Figura 10 - Rastreamento



O rastreamento que é utilizado em realidade virtual é através de dois vetores (V_x e V_y), esses dois valores são obtidos através dos vértices de interesse identificado pela leitura óptica. Quando uma dessas regiões é localizada, o primeiro teste do *software* é tentar identificar um dos cantos da região interna do marcador preto, se essa região for identificada continua o processo de mapeamento de todos os quadrados internos, porém, se não for identificado o processo é descartado.

Após o marcador ser identificado o *software* compara com valores previamente registrados, se ele for idêntico a um desses o marcador é considerado conhecido e recebe para ela a identificação correta. Para localizar a posição central do marcador, será feita a soma dos dois valores do vetor mencionado.

Após a identificação de todos os pontos feita pela leitura óptica, o *software* localiza o registro gravado e apresenta o mesmo na tela com interação 3D junto com usuário.

Outros tipos de rastreamento

– *CHROME-KEY* (Chave cromática) Esse tipo de reconhecimento utiliza o recurso de cores onde são filtradas as cores que estão no ambiente e através de sua identificação é mostrado o objeto que foi programado para a cor detectada. Uma de suas vantagens é usar um objeto arbitrário como, por exemplo, uma fichinha vermelha que ao ser identificado será apresentado um objeto sobre ela.

– *NATURAL FEATURE* (Características naturais) ainda está sendo pouco usada devido às poucas bibliotecas disponíveis, uma grande vantagem desse rastreamento é a dispensa de marcadores. Na utilização da *Natural Feature* tem como característica principal o reconhecimento de imagens a mesma abstrai os pontos que ela acha interessante e conforme o ambiente é filmado ela consegue extrair esses pontos de interesse fazendo uma analogia e detectando as bordas dessa imagem e a partir daí será passada toda a abstração para o algoritmo de estimação de pose e ele retorna a posição e orientação do objeto.

Como desenvolver a realidade aumentada

Para desenvolver a realidade aumentada será necessário bastante poder de processamento de dados, partindo da inicialização do processo de renderização dos objetos virtuais, até a sua finalização que é apresentar os objetos virtuais no mundo real, exigindo muito o processamento de dados. Essas exigências vêm da necessidade que a realidade aumentada passa a ter com o seu objetivo final que é encher os olhos dos usuários com as animações gráficas produzidas pelo computador. Para atingir tais objetivos a realidade aumentada necessita de poucos recursos de *hardware* como uma câmera, uma tela e um processador para atingir os seus resultados.

O Sistema de Realidade Aumentada com marcadores baseiam-se em marcadores confiáveis para que todos os objetos virtuais sejam renderizados nas posições corretamente. Esses marcadores devem conter uma identificação única para o ambiente que está sendo apresentado. Neste projeto será apresentado um sistema de marcador da biblioteca NyArtoolkit uma biblioteca de código aberto, desenvolvida para programação de RA.

Para construir um espaço de Realidade Aumentada o sistema busca obter imagens do ambiente real, através de imagens obtidas por uma câmera, e mistura os objetos virtuais processados por um computador com os objetos do mundo real, juntando-os no mesmo ambiente. Uma tarefa significativamente importante é extrair informações do mundo real para que o sistema de gráfico possa construir o ambiente virtual sobre a visão do usuário. Existem algumas formas de reconhecimento uma delas é o reconhecimento por marcador, que auxilia o sistema gráfico a definir coordenadas especiais e orientação dos objetos virtuais sobre qualquer movimento no mundo real.

O sistema de reconhecimento de Realidade Aumentada (RA) é projetado a partir de uma ou mais câmera que captura o ambiente real, um *software* que constrói os objetos virtuais e uma biblioteca de renderização de gráficos. Com esses três requisitos pode capturar o ambiente real com a(s) câmera(s), que passa para a biblioteca de Realidade Aumentada onde será processada a projeção de imagens virtuais e a partir dela será construído e projetado os objetos virtuais sobre os objetos

reais, e baseado nesses dados fornecido pela biblioteca de Realidade Aumentada por fim será feito o rastreamento para posicionamento e orientação dos objetos virtuais que irão interagir com os objetos do mundo real (marcador).

Reconhecimento do marcador e animação do objeto

Para se trabalhar com Realidade aumentada devemos seguir alguns passos cruciais para seu funcionamento, começando com a captura do ambiente real até a apresentação dos objetos renderizados no *display* do dispositivo. No primeiro passo são tomadas algumas medidas de grande importância para dar início a um projeto de RA. Nessa fase será processada a identificação do marcador conforme ilustrado na figura abaixo.

Figura 11 - Identificação do marcador



Fonte: <http://www.nead.ufr.br/index.php/artigos-publicados/145-capitulo-7-interacao-com-artoolkit>

Um dos pontos importantes em RA é a identificação do marcador que passa por um processo de identificação do padrão agregados aos marcadores apresentados à câmera, cada marcador tem um código único que é o código binário que fica dentro dele, esse marcado é reconhecido e a partir desse reconhecimento usamos o sistema de coordenada para podemos posicionar os objetos virtuais. Para executar a identificação o sistema aplica uma série de procedimentos na imagem apresentada como entrada para a câmera.

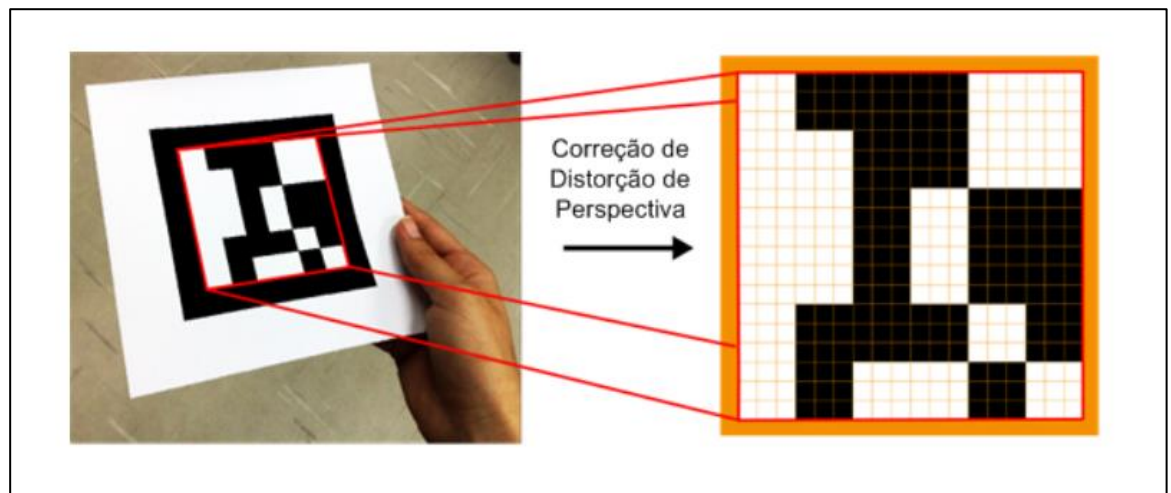
A primeira delas a ser processada é a binarização da imagem, que transforma a imagem colorida em preto e branco. Uma das grandes vantagens nesse procedimento é a quantidade de memória utilizada que será reduzida no que desrespeita ao armazenamento da imagem que será processada. Para identificar a imagem na fase de binarização utiliza-se uma limiar no valor da intensidade de cada *pixel*. Se a intensidade for maior do que a limiar, o *pixel* é considerado como branco; se a intensidade for menor do que a limiar é considerado preto.

Logo após a transformação da imagem em preto e branco, tem-se a necessidade de identificar às bordas da imagem, nesta fase de identificação a biblioteca também analisa a intensidade dos pixels onde é identificada se essa intensidade variar bruscamente. Com todas as bordas obtidas será necessário identificar os retângulos existentes na imagem, com os dados obtidos ainda temos a necessidade de identificar quais delas formam contornos fechados, executando uma classificação poligonal em todos os contornos encontrados.

A partir das informações obtidas na existência dos retângulos encontrados, temos como resultado seus vértices. A imagem ainda não está preparada para apresentação do ambiente virtual, ainda é preciso transformar a imagem para que o marcador se posicione no centro da tela. Agora será aplicado a correção da distorção de perspectiva onde será descartado toda a borda e retirado a parte inferior do quadro, ou seja, tudo que está dentro do quadrado.

Com os passos mostrados acima a biblioteca de Realidade Aumentada abstrai todo o processo de reconhecimento usando a imagem do vídeo e a partir dela te dar o sistema de coordenada que será capaz de identificar o código que está impresso no marcador, conforme mostra abaixo.

Figura 12 - Correção de distorção de perspectiva



Fonte: http://www.matmidia.mat.puc-rio.br/sibgrapi/media/undergraduate_work/59899.pdf

Renderização dos gráficos e posições e sistema de coordenada do marcador

Na renderização dos gráficos utilizamos uma biblioteca de modelagem de computação gráfica, onde é feito a modelagem e visualização tridimensional em tempo real. Toda modelagem dos objetos virtuais é feito a partir de uma biblioteca de *engine* de renderização gráfica, onde essa biblioteca nos permite a desenha um objeto virtual na tela de um computador.

Neste trabalho será aplicado a biblioteca OpenGL, uma biblioteca de computação gráfica 3D e 2D que roda em qualquer sistema operacional e que dispõe de algumas primitivas básicas (como pontos, linhas, triângulos, quadriláteros e polígonos), manipulação de sistema de coordenadas (como translação, rotação e escala) dentre outros efeitos bem como, mapeamento de textura e muitos outros comando especiais.

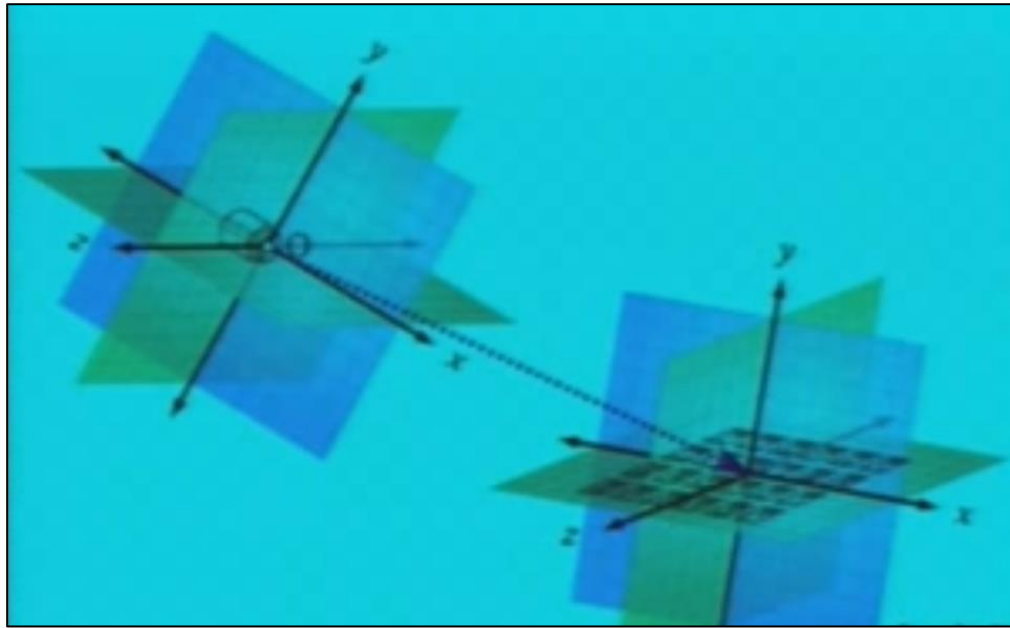
A OpenGL dispõe de algumas características especiais, uma das que se destaca entre as biblioteca de Realidade Aumentada, é basicamente a característica de sua API que atua de modo imediato. As APIs de modo imediato trabalham de

forma que quando os seus comandos são submetidos eles alteram o estado do *hardware* imediatamente em tempo real.

Na programação de Realidade Aumentada existem vários empecilhos para o programador dentre eles a renderização dos objetos. Uma saída intuitiva é o uso da biblioteca de *engine* de renderização onde não necessita de muito calculo para fazer as operações complexas como por exemplo ao invés de fazermos uma multiplicação de matriz para rotacionar um objeto, simplesmente mandamos rodar chamando uma propriedade da biblioteca que o faça o efeito que desejamos. Com o uso de biblioteca basicamente ele abstrai as operações de sua API que retorna uma operação mais imperativa ao invés de operações matemática simplificando bastante o trabalho do programador que desfruta de suas ricas propriedades ficando mais intuitivo e facilitando a vida do programador.

Para desenvolver com *engine* de renderização não basta apenas sair programando e fazendo tudo que achar certo e vier na cabeça. Para desenvolver com esse método é necessário tomar algumas medidas para que os objetos virtuais sejam desenhados de forma extremamente correto. A primeira delas é ajustar o sistema de coordenada do sistema de engine de renderização para que fique igual ao sistema de coordenada da biblioteca de realidade aumentada que é o sistema de coordenada da câmera. Na engine de renderização tem um objeto que representa a câmera, mas ele é só uma abstração onde será definido o ponto de vista dos objetos que serão desenhados e o campo de visão, então para fazer o efeito desejado, é preciso capturar o objeto da representação da câmera da engine de renderização e fazer com que ele tenha as mesmas propriedades que a câmera de verdade. Sendo assim tem a necessidade de pegar a posição da câmera que está na biblioteca de realidade aumentada, ou seja, captura a posição da câmera e traz para a origem e ajusta a posição da câmera para que ela fique alinhada com os eixos do sistema de coordenada conforme ilustrado na abaixo.

Figura 13 - Sistema de coordenada



Fonte: <http://blog.eyllo.com/tag/interatividade/>

Controle de interação e animação do objeto

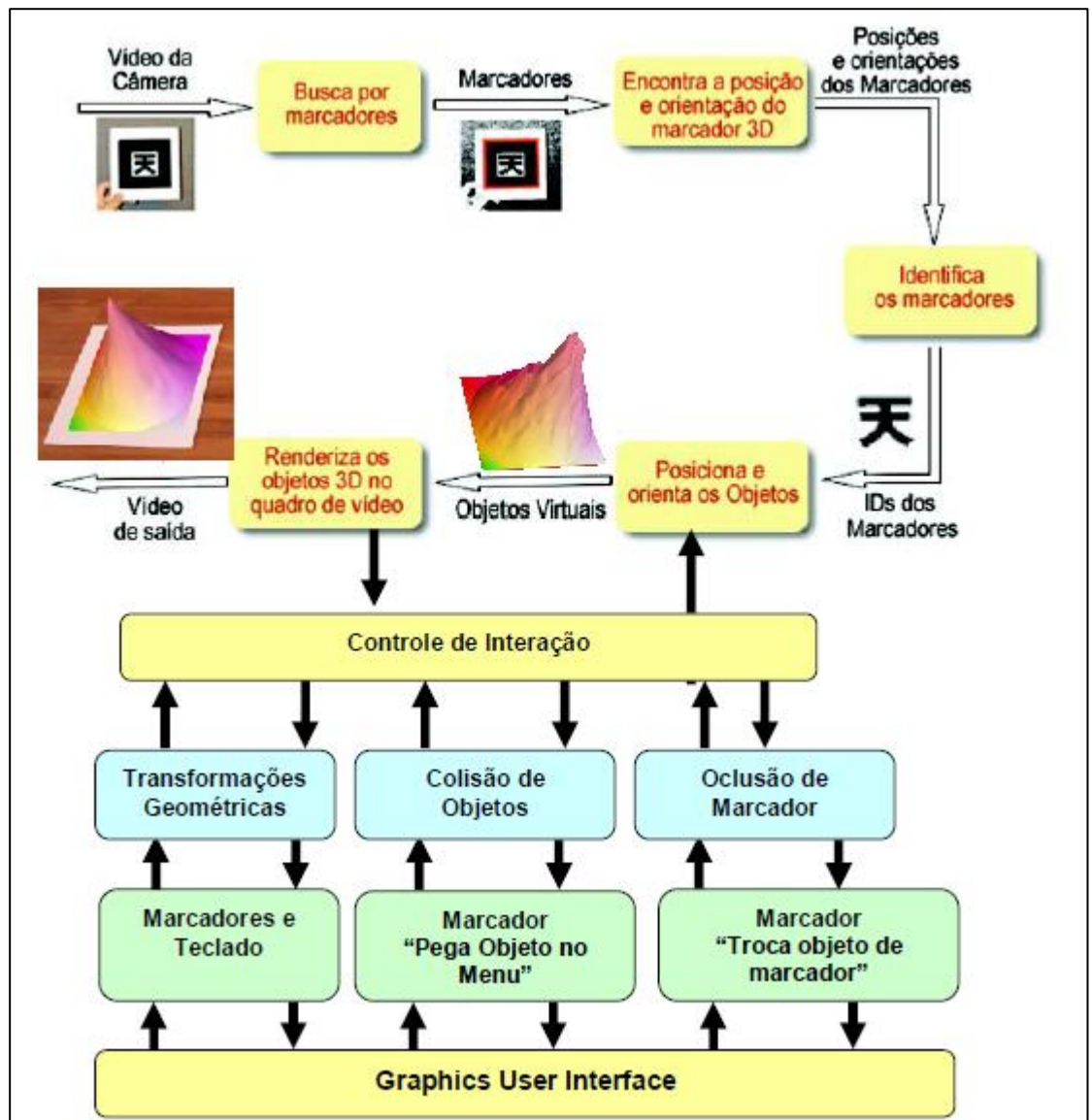
Nessa parte do projeto nos deparamos com mais algumas medidas a serem tomadas. Para apresentar um ambiente satisfatório ao usuário a biblioteca de Realidade Aumentada faz um controle de interface seguindo várias técnicas de interação. Na interface da biblioteca NyARToolkit que segue as seguintes técnicas: Primeiro São realizadas todas as transformações geométricas que estão ao seu alcance. Segundo identifica a colisão dos objetos virtuais. Para identificar a colisão a biblioteca realiza uma varredura no ambiente a procura de objetos virtuais que possam ser reconhecidos pelo *software*, esses objetos ficam em menus suspensos, onde ao colidir com o marcador ele é identificado pela biblioteca que faz a sua captura e logo após fazer a identificação o objeto virtual será instanciado sobre o marcador correspondente. Seguindo os passos anteriores a biblioteca de Realidade Aumentada chega a sua instancia final que será a oclusão de marcadores, ou seja, caso o marcador que está sendo apresentado for sobreposto por outro marcador o

sistema de Realidade Aumentada faz a leitura e verifica se o marcador que está sendo apresentado está ocluso, caso tenha obtido sucesso o marcador apresentado recebe o objeto virtual do marcador que foi sobreposto.

Graphics User Interface (GUI): Este bloco tem como característica apresentar ao usuário em modo gráfico toda apresentação gráfica em tempo real, a entrada de dados e a saída de informações. A GUI monta um ambiente de Realidade Aumentada baseado na entrada de dados e saída das informações e exibe uma janela gráfica por meio de dispositivo que tem possibilidade de capturar vídeo e mostrar através de um monitor. Essa interface gráfica permite ao usuário interagir em tempo real com o ambiente virtual. As interações em ambientes virtuais ocorrem através da possibilidade que a realidade virtual permite baseado nas transformações geométricas, interações dos objetos virtuais na apresentação, e oclusão dos marcadores.

Veja a *pipeline* completa de Realidade aumentada com NyARToolkit conforme ilustrado na figura abaixo.

Figura 14 - Interface Realidade Aumentada adaptada à Arquitetura do NyARToolkit



Fonte: <http://www.nead.ufr.br/index.php/artigos-publicados/145-capitulo-7-interacao-com-artoolkit>

Software de representação

O software foi construído totalmente em Java, utilizando o *Processing* como ambiente de desenvolvimento. O *Processing* é um ambiente de desenvolvimento integrado (IDE), que foi integrado a biblioteca de realidade Aumentada NyARToolkit, a biblioteca de renderização GSVideo com OpenLG e o kit de marcadores disponíveis *Patternmaker ARToolkit*.

Instalação

1º **Processing**: Essa IDE está disponível para *download* na seguinte URL <https://processing.org/download/?processing> , escolha a versão 1.5.1 e o sistema operacional que está rodando na máquina que será instalado. Após fazer o *download* e extrair o arquivo para um local desejado, entre na pasta e execute o arquivo executável (*Processing*) que está dentro da pasta que foi extraída.

2º **NyArtoolkit**: Para baixar a biblioteca NyArtoolkit acesse a seguinte URL http://sourceforge.jp/projects/nyartoolkit/releases/?package_id=8920, escolha a versão 1.1.6. Após fazer o *download* e extrair os arquivos para um local desejado, copie a pasta que foi extraída para o seguinte endereço do seu computador “C:\Users\usuário\Documents\Processing\libraries” esse caminho pode mudar dependendo da versão do sistema operacional e o usuário em questão é o perfil do usuário da sua máquina.

3º **GSVideo**: Para realizar o *download* a biblioteca GSVideo acesse a seguinte URL <http://sourceforge.net/projects/gsvideo/files/gsvideo/1.0/> e escolha o sistema operacional. Após fazer o *download* e extrair os arquivos para um local desejado, copie a pasta que foi extraída para o seguinte endereço do seu computador “C:\Users\usuário\Documents\Processing\libraries” esse caminho pode mudar dependendo da versão do sistema operacional e o usuário em questão é o perfil do usuário da sua máquina.

4º **Patterns**: Para que os objetos apareçam na tela do computador precisamos de um objeto real que possa ser reconhecido pelo sistema de Realidade Aumentada. A biblioteca NyARToolkit está integrado dois padrões (Hiro e Kanji), mas, não estamos limitados a isso, podemos desenvolver nossos próprios marcadores ou podemos utilizar o pacote de marcadores desenvolvido pela Universidade de Utah. Para fazer o *download* do pacote Patternmaker ARToolkit acesse a seguinte URL <http://www.cs.utah.edu/gdc/projects/augmentedreality/> . Após fazer o *download* e extrair os arquivos para um local desejado, copie a pasta que foi extraída para o seguinte endereço do seu computador

“C:\Users\usuário\Documents\Processing\libraries \nyar4psg” esse caminho pode mudar dependendo da versão do sistema operacional e o [usuário] em questão é o perfil do usuário da sua máquina.

Objetivo da aplicação

O objetivo desta aplicação é mostrar a interação do objeto virtual no mundo real sobre o reconhecimento de marcadores. O objeto apresentado é uma montanha colorida que cresce com o decorrer do tempo até um tamanho estipulado na codificação.

Codificação

```
// Software de representação do reconhecimento do QRCode usando as  
propriedades das bibliotecas.
```

```
import java.io.*; // Para a função loadPatternFileNames().
```

```
import processing.opengl.*; //Para o renderizador OPENGL
```

```
import jp.nyatla.nyar4psg.*; // Biblioteca NyARToolkit.
```

```
import codeanticode.gsvideo.*; // biblioteca GSVideo.
```

```
// Diretório do arquivo camera_para.dat
```

```
String camerapara =
```

```
"C:/Users/SERGIO.LOEIL/Documents/Processing/libraries/nyar4psg/data/camera_pa  
ra.dat";
```

```
// Diretório do arquivos que contem os marcadores . Patt

String caminhoPadrao =
"C:/Users/SERGIO.LOEIL/Documents/Processing/libraries/nyar4psg/patternMaker/ex
amples/ARToolKit_Patterns";

// Dimensões da tela . Na versão atual dessa biblioteca será possível utilizar uma
resolução máxima de 1280x720.

int largura = 640;

int altura = 360;

// Número de marcadores detectados a partir da lista completa dos arquivos Patt.
Nesse projeto será detectada os 10 primeiros da lista.

int NumMarcador = 10;

// A resolução que montanhas as montanhas serão exibidas.

int resolucaoX = 60;

int resolucaoY = 60;

// Esta é a 2 série do flutuador que todas as montanhas usam para se atualizar
durante a apresentação.

float[][] valor = new float[resolucaoX][resolucaoY];

GSCapture cam;

MultiMarker nyArtoolkit;

float[] escala = new float[NumMarcador];

float[] alteraEscala = new float[NumMarcador];

float[] alturaMontanha = new float[NumMarcador];

float[] cresceMontanha = new float[NumMarcador];
```

```
void setup() {  
  
    size(1280, 720, OPENGLE); // O esboço redimensionado na sua resolução mais alta.  
  
    cam = new GSCapture(this, 1280, 720); // Inicializar a captura da webcam em uma  
    resolução específica.  
  
    cam.start(); // Inicia a captura.  
  
    noStroke();  
  
    // Cria um novo MultiMarker com a calibração padrão da câmera e sistema de  
    coordenadas.  
  
    nyArtoolkit = new MultiMarker(this, largura, altura, camerapara,  
    NyAR4PsgConfig.CONFIG_DEFAULT);  
  
    // Defini o atraso depois que um marcador é perdido. Definido para (1), ou seja  
    imediato.  
  
    nyArtoolkit.setLostDelay(1);  
  
    // Adicionar o marcador para a detecção.  
  
    String[] padroes = loadPatternFileNames(caminhoPadrao);  
  
    // Cria uma escala individual, alteraEscala e alturaMontanha máxima para esse  
    marcador.  
  
    for (int i=0; i<NumMarcador; i++) {  
  
        nyArtoolkit.addARMarker(caminhoPadrao + "/" + padroes[i], 80);  
  
        escala[i] = random(0.8, 1.9); // Escala um pouco menor ou maior.  
  
        alteraEscala[i] = random(0.02, 0.075); // Altera a escala para torná-la bem  
        montanhoso.  
  
        alturaMontanha[i] = random(75, 150); // A altura máxima de uma montanha.  
  
    }  
  
}
```

```

void draw() {

    // Se houver uma imagem detectada pela câmara.

    if (cam.available()) {

        cam.read(); // Lê a imagem da câmara

        background(0); // chama uma imagem de fundo, necessária para a exibição
        correcta dos resultados sobre marcador.

        image(cam, 0, 0, width, height); // Exibe a imagem com a largura ea altura da
        janela do esboço.

        // Criar uma cópia da imagem da câmara na resolução de detecção AR (caso
        contrário nyArtoolkit.detect irá lançar um erro afirmação!).

        PImage cameraVirtual = cam.get();

        cameraVirtual.resize(largura, altura);

        nyArtoolkit.detect(cameraVirtual); // Detecta marcadores na imagem.

        drawMountains(); // Desenha montanhas flutuando sobre os marcadores
        detectados (3D).

    }

}

```

// Enquanto as montanhas são exibidas elas crescem (até certo ponto), caso contrário elas retornam para o estado zero.

```

void drawMountains() {

    // Define a perspectiva da biblioteca, geral para todos os marcadores.

    nyArtoolkit.setARPerspective();

    // Cria um efeito de luz na apresentação.

```

```

lights();

// loop para todos os marcadores ...

for (int i=0; i<NumMarcador; i++) {

    // Se o marcador não existe nega a execução.

    if (!InyArtoolkit.isExistMarker(i)) {

        // Se o cresceMontanha é maior do que zero, redução de 0,05, caso contrario
então continue para o próximo marcador.

        if (cresceMontanha[i] > 0) { cresceMontanha[i] -= 0.05; }

        continue;

    }

    // O código a seguir é executado se o marcador existir. Se o cresceMontanha é
inferior a 1, aumenta a montanha em 0.03.

    if (cresceMontanha[i] < 1) { cresceMontanha[i] += 0.03; }

    // O loop abaixo define os valores na 2 série do flutuador dimensional para
aumentar as montanha, com base nas variaveis (alteraEscala, alturaMontanha e no
índice (i)).

    float xoff = 0.0;

    for (int x=0; x<resolucaoX; x++) {

        xoff += alteraEscala[i];

        float yoff = 0;

        for (int y=0; y<resolucaoY; y++) {

            yoff += alteraEscala[i];

            valor[x][y] = noise(i*10+xoff+frameCount*0.05, yoff) * alturaMontanha[i]; //Define
o valor da montanha

            float distancia = dist(x, y, resolucaoX/2, resolucaoY/2);

```

```

    distancia = map(distancia, 0, resolucaoX/2, 1, 0);

    if (distancia < 0) { distancia = -distancia; } //Define os quatros campos do
    marcador.

    valor[x][y] *= distancia; // Torna o valor de zero para as extremidades
    externas.

    }

}

// Obtém a Matrix para este marcador e usa através da setMatrix.

setMatrix(nyArtoolkit.getMarkerMatrix(i));

scale(1, -1); // Virar os objetos de cabeça para baixo para trabalhar de forma
intuitiva para os usuários.

scale(escala[i]); // Escala a montanha por com a propriedade scaler

translate(-resolucaoX/2, -resolucaoY/2); // Coloca a montanha no cento do
marcador.

//Para a resolução

for (int x=0; x<resolucaoX-1; x++) {

    for (int y=0; y<resolucaoY-1; y++) {

        //Preenchimento das cores, juntos eles fazem uma montanha colorida.

        fill(x*20+y*20, 255-x*5, y*5);

        beginShape();

        vertex(x, y, valor[x][y] * cresceMontanha[i]);

        vertex((x+1), y, valor[x+1][y] * cresceMontanha[i]);

        vertex((x+1), (y+1), valor[x+1][y+1] * cresceMontanha[i]);

        vertex(x, (y+1), valor[x][y+1] * cresceMontanha[i]);

```

```
        endShape(CLOSE);  
    }  
}  
}  
  
// Redefinir a perspectiva padrão.  
perspective();  
}  
  
// Essa função carrega os arquivos .Patt em uma lista de strings baseado no  
// caminho completo do diretório.  
  
String[] loadPatternFileNames(String diretorio) {  
    File pasta = new File(diretorio);  
  
    FilenameFilter filtroPatt = new FilenameFilter() {  
        public boolean accept(File dir, String name) {  
            return name.toLowerCase().endsWith(".patt");  
        }  
    };  
  
    return pasta.list(filtroPatt);  
}
```

Conclusão

O trabalho se iniciou a partir de curiosidades sobre um tema que é pouco comum no dia a dia das pessoas (comuns/usuárias) e desenvolvedores, a mesma é apenas apresentada em filmes de ficção científica e é feita uma abordagem um tanto superficial do que a mesma é capaz de realizar, não enfatizando todos os benefícios que pode nos trazer.

Foi encontrado um vasto material com informações abrangentes sobre Realidade Virtual e Realidade Aumentada, ambas tecnologias que andam lado a lado com algumas definições e práticas um pouco diferente do que aparenta a ser, com base em tantos materiais disponíveis, decidimos apresentar os principais processos e um pequeno desenvolvimento (apresentável) de realidade virtual, como ela é trabalhada e de como ela poderá ser favorável em nosso dia a dia principalmente em trabalhos complexos como biomedicina, entretenimento, educação infantil, jogos entre outros tantos assuntos que tenha interação com as pessoas.

O trabalho foi uma tentativa de se chegar ao mais próximo do simples, o mais próximo da objetividade em que a realidade virtual é trabalhada e onde a interação com a mesma é feita.

O início do projeto foi focado no sistema que apresentamos a banca, de como o *software* se comporta com o usuário, do manuseio correto e de como ele pode ser desenvolvido e como internamente ele trabalha as imagens tridimensionais. Assim como os mais simples exemplos de como podemos trabalhar com a realidade Aumentada, mostrando alguns exemplos simples, porém, práticos para que possa ser defendido neste trabalho e que futuramente, ser desenvolvido um *software* mais completo e robusto com maior interação com pessoas, focado principalmente no público infantil.

A realidade aumentada hoje está em lojas de brinquedos, jogos de vídeo game e em até algumas apresentações profissionais e/ou no mundo artístico, onde se apresenta um artista totalmente em 3D.

A realidade Aumentada é uma tecnologia que está vindo para ficar, visando a facilitação em qualquer tipo de interação com pessoas, mostramos como a interação é de fácil manuseio por pessoas leigas e ao mesmo tempo, neste trabalho de conclusão de curso, apresentamos como suas bibliotecas de trabalho funcionam e como podem ser desenvolvidas a partir de uma documentação detalhada e elaborada.

Glossário de Siglas e Abreviaturas

2D	Duas dimensões
3D	Três dimensões
API	Interface de Programação de Aplicativos
GUI	Interface Gráfica de Usuário
IDE	Ambiente de Desenvolvimento Integrado
IHC	Interface Humano-Computador
QR Code	Código de resposta rápida (Quick Response Code)
RA	Realidade Aumentada
RV	Realidade Virtual
URL	Localizador Padrão de Recursos
VRML	Linguagem para Modelagem de Realidade Virtual

Referências Bibliográficas

FARIAS, T.; KELNER, J.; LIMA, J.; TEICHRIEB, V. **Construção de Aplicações de Realidade Aumentada Utilizando Bibliotecas de Alto Nível**. Universidade Federal de Pernambuco, Recife, 2006. Disponível em <<http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=12406>> Acessado em 02 nov. 2013.

KIRNER C.; SISCOOTTO R. **Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações** - Livro do Pré-Simpósio do SVR2007. Petrópolis, 2007. Disponível em <http://comissoes.sbc.org.br/ce-rv/home/index.php?option=com_content&view=article&id=52&Itemid=57> Acessado em 02 nov. 2013.

MACHADO, L. S.; NETTO, V. N.; OLIVEIRA, M. C. F. **Realidade Virtual: Fundamentos e Aplicações**. Florianópolis, Visual Books, 2002.

KELNER, J.; REIS, B. F.; TEICHRIEB, V.; TEIXEIRA, J. M. X. N. **Deteção de Marcadores para Realidade Aumentada**. Pernambuco, Disponível em <http://www.matmidia.mat.puc-rio.br/sibgrapi/media/undergraduate_work/59899.pdf> Acessado em 02 nov. 2013.

CARDOSO, A.; JUNIOR, E. L.; RIBEIRO, M. W.; SILVA, W. A. **Tecnologia para Desenvolvimento de Sistemas de Realidade Virtual e Aumentada: Interação com ARToolkit**. Nead UFRR. Romaima, Uberlândia, Goiás. Capítulo 7. Disponível em <<http://www.nead.ufrr.br/index.php/artigos-publicados/145-capitulo-7-interacao-com-artoolkit>> Acessado em 02 nov. 2013.

KIRNER, C.; MOREIRA, H. D. F.; SOUZA, R. C. **Flaras 1.0** - Flash Augmented Reality Authoring System. Itajubá, São Paulo, 2012. Disponível em <<http://www.ckirner.com/flaras>> Acessado em 02 nov. 2013.