UNIVERSIDADE REGIONAL DE BLUMENAU CENTRO DE CIÊNCIAS EXATAS E NATURAIS CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

UMA BIBLIOTECA DE REALIDADE AUMENTADA PARA A PLATAFORMA IOS

PAULO CESAR MEURER

PAULO CESAR MEURER

UMA BIBLIOTECA DE REALIDADE AUMENTADA PARA A PLATAFORMA IOS

Proposta de Trabalho de Conclusão de Curso submetida à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso I do curso de Ciência da Computação — Bacharelado.

Prof. Dalton Solano dos Reis – Orientador

1 INTRODUÇÃO

Atualmente, os sistemas de Realidade Aumentada (RA) se fazem presentes nos mais diversos segmentos da sociedade, tais como: publicidade, educação, comunicação e treinamento (KIRNER; TORI, 2004, p. 3-20). É notável a transformação que esta tecnologia vem produzindo no cotidiano das pessoas, permitindo a vivência de experiências muito mais ricas com os objetos e o ambiente que as cercam. A possibilidade de explorar situações diferenciadas de visualização, imersão e interação através da RA cria uma condição que favorece a atenção e a retenção de informação, simplificando a realização de tarefas outrora consideradas complexas (DAINESE; GARBIN; KIRNER, 2003, p. 273-282).

Observa-se que a crescente demanda por novas aplicações de RA e a sua potencialidade combinada com a explosão de popularidade e uso dos dispositivos móveis, cria muitas oportunidades e estimula cada vez mais os desenvolvedores a produzir sistemas para este mercado. O uso da RA em *smartphones* torna-se um meio atrativo para o usuário devido ao seu baixo custo e a ampla disponibilidade de recursos multimídia. Bajura e Neumann (1995, p. 53-60) consideram que os sistemas de RA devem ser executados em dispositivos capazes de registrar e processar as imagens com precisão, levando o usuário a crer que os mundos real e virtual ocupam o mesmo espaço. Neste contexto, o iPhone destaca-se por ser um dispositivo móvel pequeno, leve, fácil de utilizar e que reúne capacidade de processamento e recursos multimídia mínimos para este tipo de aplicação.

A robustez, padronização de bibliotecas e da estrutura de programação e um poderoso *kit* de desenvolvimento, tornam o iOS (sistema operacional do iPhone), uma plataforma ideal para a criação de sistemas de RA móvel. Porém, o desenvolvimento deste tipo de aplicação envolve a utilização e o gerenciamento de muitos recursos, como hardware, software, periféricos e redes, que exigem do desenvolvedor um conhecimento profundo de material de base, tal como computação gráfica e interface com o sistema operacional. Surge então, a necessidade de criar bibliotecas e ferramentas de desenvolvimento que promovam o reuso de estruturas e algoritmos comuns à gerência dos recursos utilizados, simplificando a criação de novas aplicações. Estas ferramentas de alto nível permitem que o desenvolvedor foque apenas no aplicativo em si, abstraindo a complexidade e reduzindo o tempo de desenvolvimento do software (GUIMARÃES; GNECCO; DAMAZIO, 2007, p. 108-128).

Diante do exposto, serão estudados os recursos disponíveis na plataforma iOS que

aplicam-se a este cenário, como o *framework AV Foundation*, o *framework* OpenGL ES¹ e a criação de bibliotecas dinâmicas dylib, para desenvolver uma biblioteca que facilite a criação de aplicações de RA utilizando a linguagem Objective-C.

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver uma biblioteca de RA em Objective-C para a plataforma iOS.

Os objetivos específicos do trabalho são:

- a) permitir o uso de recursos disponíveis pela plataforma, tais como giroscópio, Global Position System (GPS), acelerômetro, bússola e câmera digital;
- b) aplicar o conceito *markerless tracking*, utilizando coordenadas geográficas para efetuar o registro e rastreamento dos objetos virtuais;
- c) criar uma camada de abstração que oculte os detalhes de implementação da aquisição de imagens da câmera, do registro, do rastreamento, do ajuste visual dos objetos virtuais e da sobreposição das cenas do mundo real e virtual;
- d) disponibilizar um aplicativo exemplo desenvolvido utilizando a biblioteca criada.

1.2 RELEVÂNCIA DO TRABALHO

Sabe-se que os dispositivos móveis têm limitação de processamento, memória e recursos gráficos, comparados com os computadores pessoais. Estas limitações criam a necessidade de estudo e construção de algoritmos otimizados para resolver alguns problemas centrais da RA, como o alinhamento da imagem gráfica com o ambiente real, as restrições críticas de tempo para renderização das imagens virtuais e a criação de uma ilusão convincente do ambiente aumentado, onde reside a relevância deste trabalho.

Outro aspecto relevante é o uso e extensão de trabalhos já apresentados, como o de Vasselai (2010).

.

¹ Open Graphics Library for Embedded Systems (OpenGL ES).

1.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) levantamento bibliográfico: pesquisar e relacionar informações sobre assuntos abordados no trabalho, como RA, plataforma iOS e trabalhos correlatos;
- b) estudo da plataforma iOS: estudar a arquitetura e as interfaces de programação incorporadas à plataforma iOS;
- c) estudo das ferramentas correlatas: estudar algumas ferramentas disponíveis que possibilitam o desenvolvimento de RA, como o ARToolkit, Layar e Vasselai (2010);
- d) elicitação dos requisitos: reavaliar e detalhar os requisitos da aplicação, observando as necessidades identificadas durante o levantamento bibliográfico, estudo da plataforma iOS e ferramentas correlatas;
- e) especificação: especificar a biblioteca com análise orientada a objetos utilizando a *Unified Modeling Language* (UML). Será utilizada a ferramenta Enterprise Architect para a criação dos diagramas de classe, casos de uso e componentes;
- f) implementação: realizar a implementação da biblioteca proposta em linguagem Objective-C, utilizando a ferramenta Xcode e o *Software Development Kit* (SDK) do iPhone;
- g) testes: desenvolver um aplicativo exemplo para validar a biblioteca na criação de aplicações de RA.

As etapas serão realizadas nos períodos relacionados no Quadro 1.

	2011									
	jul.		ago.		set.		out.		nov.	
etapas / quinzenas	1	2	1	2	1	2	1	2	1	2
levantamento bibliográfico										
estudo da plataforma iOS										
estudo das ferramentas correlatas										
elicitação dos requisitos										
especificação										
implementação										
testes										

Quadro 1 - Cronograma

2 REVISÃO BIBLIOGRÁFICA

Na seção 2.1 é apresentado o conceito de RA e suas características. A seção 2.2 contém uma introdução à plataforma iOS, incluindo os componentes do sistema operacional, os aspectos básicos do desenvolvimento de aplicações e os principais recursos utilizados para o desenvolvimento da biblioteca proposta. Por fim, a seção 2.3 apresenta três trabalhos correlatos ao trabalho proposto.

2.1 REALIDADE AUMENTADA

Bimber e Raskar (2005, p. 2) descrevem a RA como um meio de inserir informação sintética² dentro de um ambiente real. Esta definição, apesar de genérica, é importante para que se possa compreender a diferença entre RA e Realidade Virtual (RV). Enquanto a RV imerge o usuário em um ambiente completamente artificial, a intenção da RA é manter o vínculo do usuário com a cena real permitindo que ele a perceba normalmente, apenas acrescida de novas informações virtuais (AZUMA, 1997, p. 356). O resultado desta integração entre uma cena real e estímulos gerados por computador é um ambiente híbrido, mais rico e completo, que permite ao usuário explorar situações diferenciadas de visualização e interação com os objetos e o lugar que o cercam. Porém, para que este efeito seja alcançado por um sistema de RA, o mesmo deve possuir três características básicas definidas por Azuma et al. (2001, p. 34), que são abordadas nas próximas seções.

2.1.1 Integração entre real e virtual em um ambiente real

A primeira característica descrita por Azuma (1997, p. 356) trata da combinação da realidade com a virtualidade em um cenário real. Para esta combinação, é preciso o uso de algum dispositivo de formação de imagens que permita a inserção, em tempo real, de

informações sintéticas em algum lugar no campo de visão entre o olho do observador e o ambiente real, transmitindo ao usuário a sensação de que os objetos virtuais coexistem com os reais (BIMBER; RASKAR, 2005, p. 71).

Atualmente, a maior parcela das aplicações de RA utiliza dispositivos do tipo *head-mounted* (acoplados à cabeça) para a combinação da realidade com a virtualidade. Isto se deve, em grande parte, ao fato de representarem a única possibilidade de suporte a RA móvel durante muitos anos (BIMBER; RASKAR, 2005, p. 5). Entretanto, Azuma et al. (2001, p. 35) observam que estes dispositivos constituem um fator limitante no desenvolvimento de aplicações de RA devido ao fato de não possuírem resolução, campo de visão, brilho e contraste suficientes para combinar uma grande variedade de imagens reais e virtuais.

Observando a crescente evolução tecnológica dos *tablets* e *smartphones*, Bimber e Raskar (2005, p. 5) concluem que os dispositivos *hand-held* (móveis) são a plataforma de exibição mais promissora da atualidade, pois além de permitir a combinação entre o real e o virtual, conferem ao usuário maior liberdade de movimentação, uma vez que não possuem restrição de mobilidade causada por fios e equipamentos acoplados ao corpo do usuário. Neste tipo de dispositivo, o *video see-throuth* (visão através de vídeo) é a abordagem mais utilizada. Uma câmera integrada ao dispositivo captura o fluxo de vídeo do ambiente que é então sobreposto por objetos gráficos, em tempo real, antes de ser exibido ao usuário (BIMBER; RASKAR, 2005, p. 79).

Algumas aplicações de RA, porém, não necessitam de mobilidade. Nestes casos, Bimber e Raskar (2005, p. 5) consideram que os dispositivos *spatial display* (dispositivo espacial) são muito mais eficientes, pois conseguem alcançar um nível de qualidade e realismo maior do que qualquer dispositivo móvel (*hand-held* ou *head-mounted*) da atualidade.

2.1.2 Interação em tempo real

A segunda característica descrita por Azuma (1997, p. 356) trata da interatividade em tempo real. Uma vez que a realidade aumentada concentra-se em sobrepor o ambiente real por elementos gráficos, os mesmos devem ser capazes de criar uma ilusão convincente de

² Informação sintética refere-se a um estímulo virtual dos sentidos. Pode ser a simulação de uma imagem, um

realidade, incorporando as sombras do ambiente e o comportamento de inter-reflexão dos demais objetos. A manutenção de uma ilusão realista é muito importante para que seja mantida a idéia de continuidade entre o mundo real e virtual e, para tanto, métodos para renderização em tempo real têm um importante papel (BIMBER; RASKAR, 2005, p. 5).

Segundo Fournier (1994, p. 2), existem dois tipos de problemas de realismo na RA: os geométricos e os de iluminação. Os problemas geométricos residem na computação de parâmetros de visualização, que remetem ao problema do registro apresentado na seção 2.1.3, e na determinação da visibilidade dos objetos virtuais. Já os problemas de iluminação consistem na iluminação de objetos reais com fontes de luz geradas por computador, iluminação de objetos virtuais com fontes de luz real e problemas considerados secundários como reflexos, transparências e sombras.

2.1.3 Registro e rastreamento dos objetos em terceira dimensão

A terceira e última característica descrita por Azuma (1997, p. 356) trata do registro dos objetos virtuais em terceira dimensão. Entende-se registro como o alinhamento dos objetos virtuais com a cena real através de algum mecanismo de rastreamento que pode ser ótico, mecânico, magnético, inercial ou ultrassônico.

Bimber e Raskar (2005, p. 4) consideram que o sucesso de uma aplicação de RA depende do registro e rastreamento preciso, rápido e robusto da posição dos objetos reais e virtuais em relação ao usuário e seu ponto de vista, pois o desalinhamento entre estes objetos pode causar a perda de noção da realidade.

De acordo com Cawood e Fiala (2007, p. 4), existem vários métodos para registrar a posição dos objetos virtuais no ambiente real, que vão desde a utilização de *Light-Emitting Diodes* (LEDs) até a mão de uma pessoa. Entretanto, o modo mais simples de registro é conhecido como *marker-based tracking*. Este método consiste em definir um marcador que possui um padrão único e adicioná-lo fisicamente no ambiente real. Quando este padrão é reconhecido pelo sistema de RA, é possível determinar a posição e o ângulo do marcador e então utilizar estas informações para exibir o objeto virtual na posição e orientação corretas.

O uso de marcadores, porém, não é efetivo quando o sistema de RA é executado em

grandes áreas abertas. Nestes ambientes o registro é comumente feito sem o uso de marcadores, utilizando um método conhecido como *markerless tracking*. Cawood e Fiala (2007, p. 6) consideram este método de registro e rastreamento como sendo o ideal, pois não exige a adição prévia de marcadores no cenário. Para Bimber e Raskar (2005, p. 5), esta é a solução mais promissora para o futuro das aplicações de RA, porém, mesmo contando com o apoio de tecnologias como o GPS para registrar os objetos e dispositivos de medição relativa como o giroscópio e o acelerômetro para determinar a posição e orientação corretas, é a solução mais desafiadora.

2.2 PLATAFORMA IOS

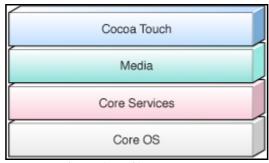
O iOS é uma plataforma de software que abrange o sistema operacional e um conjunto de tecnologias que são utilizadas para executar aplicativos nativamente nos dispositivos móveis da empresa Apple, como o iPad, iPhone e iPod Touch. Embora sua estrutura herde muitas tecnologias da plataforma Mac OS X, o iOS possui funcionalidades adicionais, como por exemplo a interface *Multi-Touch* e suporte ao acelerômetro, para satisfazer as necessidades de um ambiente móvel, onde as necessidades dos usuários são ligeiramente diferentes (APPLE INC, 2010a).

2.2.1 Arquitetura do sistema operacional

Nas camadas mais baixas, a arquitetura do iOS é semelhante à arquitetura básica do Mac OS X. Estes componentes compartilhados entre as duas plataformas são conhecidos coletivamente como Darwin (RIDEOUT, 2010, p. 1).

Em seu nível mais alto, porém, o iOS possui um conjunto de tecnologias que age como um intermediário entre o *hardware* e os aplicativos que aparecem na tela. A implementação destas tecnologias pode ser vista como um conjunto de quatro camadas: *Cocoa Touch*, *Media*, *Core Services* e *Core OS*, como observado na Figura 1. A abstração fornecida por estas camadas faz com que as aplicações dificilmente comuniquem-se diretamente com o *hardware*, permitindo que funcionem de forma consistente mesmo em dispositivos com

recursos de hardware diferentes (APPLE INC, 2010b).



Fonte: Apple Inc (2010b).

Figura 1 — Visão geral das camadas de alto nível da plataforma iOS

Nas duas camadas inferiores estão os serviços fundamentais e as tecnologias de que todos os aplicativos dependem, como o acesso a arquivos, tipos de dados de baixo nível e *sockets* de redes, por exemplo. Já as duas camadas superiores fornecem abstrações orientadas a objetos das camadas de nível inferior e disponibilizam serviços e tecnologias mais sofisticados.

A camada *Core OS* contém os recursos de baixo nível sobre os quais a maior parte das outras tecnologias é construída. Geralmente é acessada pelas outras camadas, a não ser em casos onde há necessidade de lidar explicitamente com questões de segurança ou efetuar comunicação com algum equipamento externo. Esta camada lida basicamente com acesso ao sistema de arquivos, entrada e saída padrão de dados, alocação de memória, computações matemáticas e segurança dos dados manipulados pelas aplicações (APPLE INC, 2010f).

A camada *Core Services* oferece os serviços fundamentais que são utilizados em todas as aplicações, como por exemplo, o *framework CFNetwork* que oferece suporte para trabalhar com os protocolos de rede, o *framework Core Data* que permite utilizar ferramentas gráficas para construir e gerenciar o modelo de dados de uma aplicação e o *framework Foundation* que oferece suporte às coleções de dados, pacotes, gerenciamento de preferências, *strings*, data, hora, *threads* e etc. (APPLE INC, 2010e).

A camada *Media* contém os *frameworks* necessários para acessar a maioria dos protocolos de áudio e vídeo. Dentre estes, o *framework AV Foundation* oferece serviços como edição de mídia, reprodução de áudio e vídeo, gerência de metadados para itens de mídia e a captura de vídeo. Esta camada ainda oferece suporte a desenhos 2D e 3D através de tecnologias gráficas como o OpenGL ES, o *Embedded Apple Graphics Library* (EAGL), o *Quartz* e o *Core Animation* (APPLE INC, 2010d).

A camada *Cocoa Touch* contém os *frameworks* chave para a construção de aplicações no iOS. Nela são definidas a infra-estrutura básica das aplicações e o suporte para tecnologias

fundamentais, tais como multitarefa, entrada baseada em toque, notificações *push*, reconhecedores de gestos e diversos serviços de alto nível. O *framework* UIKit, por exemplo, disponibiliza recursos em interfaces gráficas, como as classes de visões, controles, janelas e os controladores destes objetos (APPLE INC, 2010c).

2.2.2 OpenGL ES

O OpenGL ES é uma versão simplificada do OpenGL que elimina algumas funcionalidades redundantes, com a finalidade de fornecer uma *Application Programming Interface* (API) reduzida para uso em dispositivos embarcados como os *smartphones*.

Por ser uma API baseada na linguagem C, o OpenGL ES é extremamente portável e se integra perfeitamente com aplicações *Cocoa Touch* desenvolvidas com Objective-C. Porém, a especificação do OpenGL ES não define uma camada de janelas, em vez disso, o sistema operacional deve disponibilizar funções para criar um contexto de renderização do OpenGL que aceita comandos e um *framebuffer* onde os resultados dos comandos de desenho são escritos. Para tanto, o iOS disponibiliza o *framework* EAGL, que age como uma interface entre o OpenGL e o UIKit (APPLE INC, 2011).

Um detalhe importante a ser observado quando se utiliza o OpenGL ES no desenvolvimento de um aplicativo, é que apesar de possuir um poderoso sistema *multitasking*, o iOS não permite que aplicações que estejam executando em *background* chamem as suas funções. Isto é feito para que o processador gráfico esteja sempre completamente disponível para a aplicação que executa em *foreground*. Quando um aplicativo acessa o processador gráfico enquanto está em *background*, ele é automaticamente encerrado pelo iOS. Esta restrição não se aplica somente a novas chamadas de funções do OpenGL, mas também a comandos que foram previamente submetidos e não tenham sido completados. Portanto, as aplicações que utilizam o OpenGL ES devem garantir que todos os comandos previamente submetidos sejam concluídos antes de mudar o seu estado para o *background* (APPLE INC, 2011).

2.2.3 Bibliotecas dinâmicas .dylib

Grande parte das funcionalidades de uma aplicação provém de bibliotecas de código executável implementadas pelos sistemas operacionais. Durante o desenvolvimento de uma aplicação, o desenvolvedor deve ligar o código de sua aplicação com estas bibliotecas para obter acesso a estas funcionalidades. Esta ligação, quando ocorre de maneira estática, faz com que código de biblioteca que o aplicativo utiliza seja copiado para o arquivo executável gerado e carregado na memória durante toda a sua execução.

Segundo Apple Inc (2009), dois fatores importantes para que os aplicativos apresentem boa performance são o tempo gasto na inicialização e o uso correto da memória. Estes requisitos podem ser alcançados reduzindo o tamanho do arquivo executável e o uso de memória pelas aplicações. Uma solução que atende a estas necessidades é o uso de bibliotecas dinâmicas, que permite às aplicações retardarem o carregamento de bibliotecas de funcionalidades especiais até o momento em que elas são necessárias, ao invés de carregar a biblioteca inteira durante sua inicialização. As aplicações escritas para o iOS são beneficiadas por este recurso, pois todas as bibliotecas de sistema, os arquivos .dylib, no Mac OS X são dinâmicas.

2.3 TRABALHOS CORRELATOS

Existem várias ferramentas para a RA que auxiliam na construção dos objetos virtuais e em sua integração com o ambiente real. Para o uso em computadores pessoais, pode-se citar o ARToolkit (LAMB, 2010). Nas plataformas móveis, dentre outras, pode-se destacar ferramentas como o Layar (LAYAR, 2010) e Vasselai (2010).

O ARToolkit é um *software* que utiliza marcadores para o rastreamento e registro de objetos virtuais no ambiente real. Estes marcadores são expressos por padrões 2D na forma de um quadrado com bordas pretas e são previamente cadastrados e inseridos no ambiente real antes da execução da aplicação. A detecção destes marcadores é feita analisando a imagem com o auxílio de técnicas de visão computacional. Segundo Lamb (2010), as seguintes etapas são executadas para detectar os marcadores: a imagem capturada é transformada em uma imagem binária, logo após, o *software* analisa a imagem e detecta os marcadores, e os

compara com os previamente cadastrados. Quando um marcador é encontrado, um objeto virtual é adicionado à imagem real, na posição e orientação do marcador original.

Layar é um *framework* que permite o desenvolvimento de aplicativos com realidade aumentada através de um servidor próprio (LAYAR, 2010). Os aplicativos criados com o Layar utilizam GPS, bússola e câmera para identificar o ambiente real e mostrar, em tempo real, informação digital no campo de visão do usuário através do seu dispositivo móvel. Conforme Layar (2010), o Layar destaca-se por ser uma plataforma que suporta um nível sofisticado de interatividade, incluindo elementos de áudio e vídeo, modelos 3D e capacidades de compartilhamento social.

Vasselai (2010) descreve uma aplicação de RA desenvolvida para a plataforma Android que apresenta características semelhantes às propostas por este trabalho, dentre as quais se podem destacar:

- a) os objetos virtuais são registrados através de coordenada geográfica;
- b) utiliza a câmera do dispositivo para capturar as imagens reais;
- c) permite selecionar os objetos virtuais através do multitoque;
- d) utiliza o GPS e o acelerômetro do dispositivo;
- e) os pontos de interesse podem ser objetos 2D;
- f) possui um serviço cadastro dos pontos de interesse.

3 REQUISITOS DO SISTEMA A SER DESENVOLVIDO

Os requisitos para a biblioteca proposta são:

- a) permitir visualizar o ambiente real através da câmera do dispositivo (Requisito Funcional - RF);
- b) sobrepor ao ambiente real com objetos virtuais em 2D (RF);
- c) utilizar o multitoque para visualizar detalhes dos objetos virtuais (RF);
- d) utilizar o conceito de *markerless tracking* para registro dos objetos virtuais através de coordenada geográfica (RF);
- e) utilizar o acelerômetro para rastrear a direção que o usuário está visualizando com o dispositivo (Requisito Não Funcional - RNF);
- f) utilizar o giroscópio para identificar o ângulo de visão do usuário em relação ao objeto virtual (RNF);
- g) obter a coordenada geográfica e informações dos objetos virtuais através da rede 3G e/ou wireless (RNF);
- h) ser implementado na plataforma iOS.

4 CONSIDERAÇÕES FINAIS

A crescente demanda por novas aplicações de RA e a sua potencialidade combinada com a explosão de popularidade e uso dos dispositivos móveis, cria muitas oportunidades e estimula cada vez mais os desenvolvedores a produzir sistemas para este mercado. O iPhone destaca-se no campo da RA móvel como um dispositivo de exibição que reúne capacidade de processamento de vídeo significativa e recursos como GPS, acelerômetro, giroscópio, *Multi-Touch* e bússola em um único dispositivo. A plataforma iOS mostra-se um excelente ambiente para a criação de aplicações de RA, fornecendo mecanismos para controlar estes recursos, um *kit* de desenvolvimento robusto e uma estrutura de programação e de bibliotecas padronizada e fácil de utilizar.

Com relação aos trabalhos correlatos apresentados, pode-se verificar que possuem funcionalidades comuns às que estão sendo propostas por este trabalho. Layar é um framework multiplataforma que possui muitas semelhanças a este trabalho, sendo a mais relevante o fato de não utilizar marcadores para o registro dos objetos virtuais. O registro é feito utilizando coordenadas geográficas e o acelerômetro. Como diferencial, o trabalho proposto utiliza o giroscópio como tentativa de amenizar o problema do realismo geométrico. Vasselai (2010), apesar de servir como base para este trabalho, descreve uma aplicação totalmente voltada para a plataforma Android, utilizando recursos exclusivos da mesma como o paradigma de intenções. Neste sentido, o trabalho proposto se difere completamente de Vasselai (2010), pois será desenvolvido para outra arquitetura de software.

Pretende-se com o trabalho proposto criar uma biblioteca de RA baseada nas funcionalidades apresentadas em Vasselai (2010), que abstraia os detalhes de implementação das características básicas de uma aplicação de RA móvel para a plataforma iOS. Permitindo aos desenvolvedores focar o seu trabalho no desenvolvimento da aplicação em si.

REFERÊNCIAS BIBLIOGRÁFICAS

APPLE INC. iOS overview. [S.l.], 2010a. Disponível em: http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/URL_iPhone_OS_Overview/index.html . Acesso em: 8 mar. 2011.
About iOS development. [S.l.], 2010b. Disponível em: <a #documentation="" 0007898-ch3-sw1="" apple_ref="" conceptual="" developer.apple.com="" doc="" href="http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOS TechOverview/IPhoneOSOverview/IPhoneOSOverview.html#//apple_ref/doc/uid/TP40007898-CH4-SW1>. Acesso em: 8 mar. 2011.</td></tr><tr><td> Cocoa touch layer. [S.1.], 2010c. Disponível em: . Acesso em: 8 mar. 2011.
Media layer. [S.l.], 2010d. Disponível em: <a #documentation="" conceptual="" developer.apple.com="" href="http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOS TechOverview/MediaLayer/MediaLayer.html#//apple_ref/doc/uid/TP40007898-CH9-SW4>. Acesso em: 8 mar. 2011.</td></tr><tr><td> Core services layer. [S.1.], 2010e. Disponível em: http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOS TechOverview/CoreServicesLayer/CoreServicesLayer.html#//apple_ref/doc/uid/TP40007898-CH10-SW5>. Acesso em: 8 mar. 2011.
Core os layer. [S.l.], 2010f. Disponível em: <a #documentation="" 3ddrawing="" apple_ref="" conceptual="" developer.apple.com="" doc="" href="http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptual/iPhoneOS TechOverview/CoreOSLayer/CoreOSLayer.html#//apple_ref/doc/uid/TP40007898-CH11-SW1>. Acesso em: 8 mar. 2011.</td></tr><tr><td> About OpenGL ES. [S.l.], 2011. Disponível em: . Acesso em: 8 mar. 2011.
Overview of dynamic libraries. [S.l.], 2009. Disponível em:

AZUMA, Ronald T. A Survey of augmented reality. **Presence**: Teleoperators And Virtual Environments, [S.l.], v. 6, n. 4, p. 355-385. Aug. 1997. Disponível em: http://www.cs.unc.edu/~azuma/ARpresence.pdf>. Acesso em: 25 fev. 2011.

AZUMA, Ronald T et al. Recent advances in augmented reality. **IEEE Computer Graphics & Applications**, Los Alamitos, v. 21, n. 6, p. 34-47, Nov. 2001.

BAJURA, Michael; NEUMANN, Ulrich. Dynamic registration correction in video-based augmented reality systems. **IEEE Computer Graphics & Applications**, Los Alamitos, v. 15, n. 5, p. 53-60, Sept. 1995.

BIMBER, Oliver; RASKAR, Ramesh. **Spatial augmented reality merging real and virtual worlds**. Wellesley: A K Peters, 2005.

CAWOOD, Stephen; FIALA, Mark. **Augmented reality**: a practical guide. Raleigh: The Pragmatic Bookshelf, 2007.

DAINESE, Carlos A.; GARBIN, Tânia R.; KIRNER, Claudio. Sistema de realidade aumentada para desenvolvimento cognitivo da criança surda. In VIDAL, Creto A.; KIRNER, Claudio. **Proceedings of 6th svr symposium on virtual reality**. Ribeirão Preto: SBC, 2003. p. 273-282.

FOURNIER, Alain. Illumination problems in computer augmented reality. In: Journee INRIA Analyse-Synthèse d'Images, 1., 1994, Paris. **Actes Texte Impremé**... Paris: INRIA, 1994. p. 1-21. Disponível em: http://www.cs.ubc.ca/labs/imager/tr/ps/fournier.19. Acesso em: 3 mar. 2011.

GUIMARÃES, Marcelo P.; GNECCO, Bruno B.; DAMAZIO, Rodrigo. Ferramentas para desenvolvimento de aplicações de realidade virtual e aumentada. In: KIRNER, Claudio; SISCOUTTO, Robson. **Realidade virtual e aumentada**: conceitos, projetos e aplicações. Petrópolis: SBC, 2007. p. 108-128.

KIRNER, Claudio; TORI, Romero. Introdução à realidade virtual, realidade misturada e hiper-realidade. In: KIRNER, Claudio; TORI, Romero. **Realidade virtual:** conceitos, tecnologia e tendências. São Paulo: SENAC, 2004. p. 3-20.

LAMB, Philip. **ARToolkit**. Washington, 2010. Disponível em: http://www.hitl.washington.edu/artoolkit/>. Acesso em: 14 mar. 2011.

LAYAR. **Layar for iPhone**. Amsterdam, 2010. Disponível em: http://www.layar.com/download/iphone/>. Acesso em: 14 mar. 2011.

RIDEOUT, Philip. **iPhone 3D programming**. Sebastopol: O'Reilly Media, 2010.

VASSELAI, Gabriela T. **Um estudo sobre realidade aumentada para plataforma Android**. 2010. 103 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências exatas e Naturais, Universidade Regional de Blumenau, Blumenau.