

UMA PROPOSTA DE MIDDLEWARE PARA DISTRIBUIÇÃO DE REALIDADE AUMENTADA PELA WEB COMO FERRAMENTA DE APOIO PARA EDUCAÇÃO A DISTÂNCIA

Marlus Dias Silva, Eduardo Souza Santos, Webert Vieira Arantes,
Lázaro Vinícius de Oliveira Lima, Alexandre Cardoso, Edgard Lamounier

Universidade Federal de Uberlândia
Uberlândia, MG, Brasil

{marlusdias, eduardo}@mestrado.ufu.br, webert3d@yahoo.com.br,
lazavini@gmail.com, {alexandre, lamounier}@ufu.br

Abstract

In this work, the development of a Middleware used to distribute Augmented Reality in the web, as an auxiliary tool to Distance Education is presented. The software is divided in three modules: server, teacher and student. In the teacher module, Augmented Reality concepts can be used in teaching, introducing virtual objects in the real environment, by the use of markers. In the student module, students can visualize teacher's interactions and communicate by chat. Finally, the server module provides connection management and data distribution, from the teacher to the students. The Middleware was developed using C#.net and Action Script 3.0 programming languages, supported by the Flex 2.0 development environment.

1 Introdução

Com o advento da Educação a Distância (EaD), a necessidade de prover ambientes que facilitem o aprendizado se mostra ainda mais importante, já que alunos dessa modalidade de aprendizado possuem poucas ferramentas que possam estimular a experimentação, dificultando seu aprendizado e, portanto, causando defasagens em relação à educação de um aluno da modalidade presencial [11].

Neste contexto, observa-se que a utilização da Realidade Aumentada (RA), que é definida como uma fusão entre o mundo real e objetos virtuais, criando um cenário no qual as informações contidas no mundo real são incrementadas com objetos virtuais gerados por computador [3], pode oferecer informações sensíveis mais ricas, facilitando a associação e a reflexão sobre o tema que está sendo ensinado [6], possibilitando que o usuário possa lidar de forma mais con-

fortável com conceitos abstratos e contra-intuitivos [7].

Segundo [12], um Sistema distribuído “é uma coleção de computadores independentes que se apresentam aos usuários como um sistema único e coerente”, dessa forma a possibilidade de distribuir RA e RV torna-se muito interessante pois desta forma vários usuários em diversas posições geográficas podem interagir com o sistema em tempo real, favorecendo a colaboração, o que facilita de forma expressiva o aprendizado.

Dessa forma, é de suma importância buscar tecnologias que permitam a distribuição de ambientes virtuais, que estejam alinhadas com as atuais necessidades tecnológicas. Nesse âmbito, pode-se destacar as aplicações desenvolvidas no ambiente *Flex 2.0*, que tem se desenvolvido de forma muito rápida e com uma ótima aceitação tanto por desenvolvedores quanto no mercado consumidor.

O lançamento do *ActionScript 3.0* (AS3), juntamente com o ambiente de desenvolvimento *Flex 2.0* e o *Flash Player 9.0*, pela Adobe em 2006, abriu caminho para a evolução das aplicações RIA (*Rich Internet Applications* - Aplicações Ricas em Internet). O desenvolvimento dessas aplicações possibilitou a criação de diversas ferramentas de apoio para a criação de aplicações de Realidade Virtual (RV) para a *web*, suportadas pelo *Flash Player*. Como passo seguinte, a extensão destas tecnologias proveu condições de desenvolvimento de soluções de Realidade Aumentada (RA) para *web*, com a utilização de uma *webcam* e marcadores (impressos em papel comum) e da tecnologia FLARToolkit.

Neste contexto, este trabalho tem como objetivo mostrar uma proposta de desenvolvimento de um *middleware* para a distribuição de objetos com de Realidade Aumentada (RA) na *web*, permitindo que vários usuários, conectados a um servidor, possam assistir a uma apresentação na qual é uti-

lizada RA, em tempo real, havendo necessidade da utilização de somente uma *webcam*, que deve estar conectada ao computador do apresentador.

Além da possibilidade de criar aplicações de RA na *web*, o ambiente de desenvolvimento *Flex 2.0* oportuniza a utilização das aplicações desenvolvidas em qualquer plataforma, sendo transparente para o usuário, já que não se necessita da instalação da aplicação. Logo, essa proposta utiliza tal ambiente para o desenvolvimento do *front-end*, possibilitando que o aluno/professor, utilize as aplicações em qualquer navegador que possua o plugin *Flash Player* com a versão 9 ou superior instalado, permitindo sua utilização em sistemas operacionais Windows, Linux, Solaris ou Macintosh. Além da vasta gama de sistemas operacionais, observa-se que o *Flash Player 9*, possui uma taxa de utilização de aproximadamente 99% nos computadores dos mercados maduros (Estados Unidos, Reino Unido, Alemanha, França e Japão) e cerca de 97% nos mercados emergentes [1].

Com o intuito de abordar o desenvolvimento e utilização do *middleware* desenvolvido, este trabalho será dividido da seguinte forma: a seção 2, trata dos trabalhos correlatos, mostrando a importância da utilização de RV e RA na educação, bem como ferramentas de distribuição dessas tecnologias. A terceira seção relata as ferramentas utilizadas no desenvolvimento da arquitetura proposta. A seção número 4, mostra as definições da arquitetura proposta e, por fim, na quinta sessão são mostrados os resultados e perspectivas para trabalhos futuros.

2 Trabalhos Relacionados

Devido às possibilidades de aplicação no ensino, a distribuição de objetos virtuais para ambientes de Realidade Aumentada, tem sido bastante explorada. O trabalho de [5], por exemplo, apresenta uma interface para a distribuição de objetos virtuais utilizando realidade aumentada como ferramenta de apoio de ensino à distância na área fisiologia vegetal. A arquitetura utilizada foi cliente servidor utilizando o CORBA (*Common Object Request Broker Architecture*) para a comunicação entre os clientes que executam o ARToolkit.

Em [6] uma ferramenta de distribuição de ambientes virtuais para apoio a projetos multidisciplinares de ensino. Foram desenvolvidos dois protótipos: um na área biológica e outro na área de química. Para a distribuição foi utilizado o CORBA, utilizando arquitetura cliente/servidor.

De forma semelhante aos trabalhos supracitados, este trabalho utiliza a arquitetura cliente/servidor, no entanto, com a utilização de tecnologias diferentes, visto que deseje-se distribuir o ambiente de Realidade Aumentada, utilizando o ambiente de desenvolvimento *Flex*, permitindo que os clientes possam utilizar a ferramenta desenvolvida

de forma transparente, necessitando somente de uma *webcam* e um navegador com o *Flash Player* instalado.

3 Ferramentas utilizadas

As ferramentas utilizadas para o desenvolvimento da arquitetura são apresentados nas seções abaixo :

3.1 Engine 3D

Atualmente existem diversas *engines* para desenvolvimentos de ambientes 3D em AS3, como *Away3D*, *Papervision3D* e *Sandy*, logo, é de suma importância definir qual desses conjuntos de bibliotecas serão utilizados no projeto, visto que, dependendo do tipo de aplicação que será desenvolvida, a má escolha da *engine* pode causar um desempenho ruim do sistema como um todo.

Por este motivo, baseado no grau de atividade e avaliação das *engines* pelos desenvolvedores, optou-se por utilizar o *Papervision3D*, que surgiu em 2005 e, desde então, vem sendo atualizado pelos desenvolvedores e pela vasta comunidade de utilizadores.

Essa engine possui um conjunto de classes que possibilita o tratamento de colisão de forma automatizada, *engines* para a simulação de física (simulação de fluidos e partículas) possibilitando gerar mundos virtuais com uma maior riqueza de detalhes, aproveitando todos os benefícios da Orientação a Objetos. E, além disso, torna possível a importação de objetos modelados em ambientes de modelagem 3D, como Blender e 3D Studio Max, e exportados nos formatos Collada (*COLLABorative Designer Activity*), MD2(*Quake 2*) e 3DS(*3D Studio*).

Outro ponto de destaque, consiste na integração com outra biblioteca AS3, chamada FLARToolkit, que é semelhante ao ARToolkit e, portanto, possibilita o desenvolvimento de aplicações de Realidade Aumentada.

3.2 FLARToolkit

O *FLARToolkit* é um conjunto de classes desenvolvidas em AS3, que juntamente com o *Papervision3D* possibilita o desenvolvimento de aplicações de Realidade Aumentada. Para executar aplicações utilizando o *FLARToolkit* o cliente deve instalar no navegador o *Flash Player* versão 9 ou superior, e permitir o acesso do *plugin* à *webcam*, através do *pop-up* mostrado na Figura 1.

O *ARToolKit* é uma biblioteca em linguagem C que permite aos programadores desenvolver aplicações de Realidade Aumentada [4]. Desta forma, para executar aplicações utilizando o *ARToolkit* o usuário deverá ter instalado e configurado o mesmo em seu computador, podendo causar algum desgaste para usuários menos experientes. Em contrapartida, uma aplicação que utiliza o *FLARToolkit* será



Figura 1. Flash Player solicitando acesso a webcam.

disponibilizada como um arquivo “.swf”, que é executado no *Flash Player* do navegador *web*.

3.3 FluorineFX

O *FlourineFX* é um *Middleware* de código fonte aberto que possibilita a integração do *Flash/Flex* com a linguagem *Microsoft .NET Framework* através de *RPC (Remote Procedure Call RPC)* em tempo real [8].

Chamada Remota a Procedimento (RPC) permite que programas invoquem procedimentos ou funções localizados em outras máquinas, como essas se estivessem sendo executadas localmente. A nível do programa, as informações são passadas do chamador para o procedimento chamado através dos parâmetros, os resultados são retornados através dos procedimento. Possibilitando ao programador um maior nível de abstração no desenvolvimento de suas aplicações, onde não precisa-se preocupar com a conexão tanto para redes locais como para redes externas [9, 2].

Logo, o *FlourineFX* foi escolhido por possibilitar a utilização com aplicações desenvolvidas em *Flex* e suportar as vantagens oferecidas por RCP.

4 Arquitetura do sistema

A arquitetura foi dividida em três módulos: aluno, servidor e professor. Cada módulo possui sua particularidade como mostra a Figura 2.

4.1 Módulo Professor

Esse módulo é responsável pela captura das imagens, que são geradas conforme ocorre a mudança do posicionamento do marcador. Posteriormente, essas informações são convertidas em um *Array* de Bytes que é enviado ao módulo servidor, através de uma chamada remota a procedimento, com o auxílio ao *FlourineFX*.

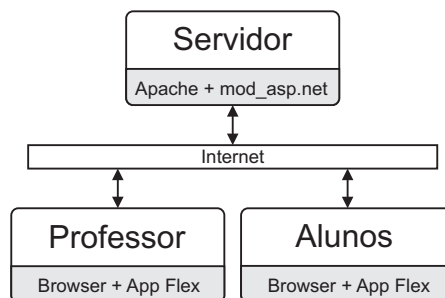


Figura 2. Arquitetura do Sistema

Para o desenvolvimento desse módulo foi utilizado a linguagem AS3, no ambiente *Flex*, que possibilita a integração com o *FlourineFX*. A figura 3 mostra a interface visual deste módulo, no painel a direita observa-se o vídeo que é gerado com elementos de Realidade Aumentada.

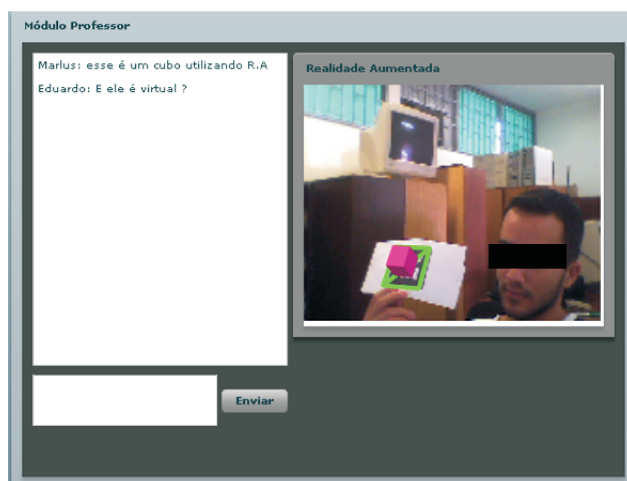


Figura 3. Interface do Módulo Professor

O professor/apresentador pode determinar ações sobre o módulo como mostra a Figura 4. Este após permitir o acesso à *webcam*, pode começar a transmitir os dados, com elementos de Realidade Aumentada, podendo parar e reiniciar a transmissão para o Servidor a qualquer momento.

Além disso, esse módulo contém um *chat*, no qual o Docente pode comunicar-se com os alunos, para que todos participem ativamente da aula.

4.2 Módulo Servidor

Este módulo é responsável por gerenciar as conexões com os usuários do sistema e receber/distribuir os dados que são distribuídos. O *FlourineFX* age diretamente no servidor, visto que é o *Middleware* responsável pela RPC que, através de sua interface remota, envia os dados para os clientes que solicitam o serviço.

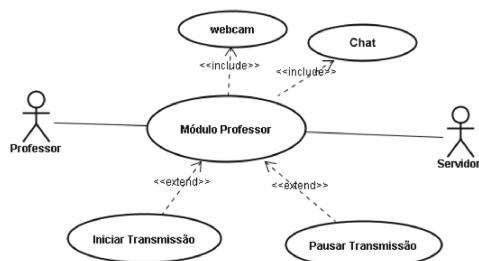


Figura 4. Caso de Uso Professor

Como a interface do servidor foi desenvolvida utilizando a linguagem C#, o módulo Servidor necessita de uma infraestrutura com suporte à linguagem Asp.NET instalada. Tal requisito, consiste em um servidor IIS(*Internet Information Services*) ou Apache com suporte a .NET, possibilitando a execução de aplicações .NET na *web* [10].

4.3 Módulo Aluno

Esse módulo é responsável por requisitar as informações geradas pelo Módulo Professor e armazenadas no Servidor, através da RPC. Dessa forma, os dados gerados no Módulo Professor que são atualizados nos clientes, possibilitando a visualização as ações efetuadas pelo apresentador. A Figura 5 mostra a interface do cliente. Pode-se observar que os clientes não podem enviar imagens, ou seja, só veem o que está acontecendo do lado do módulo professor.

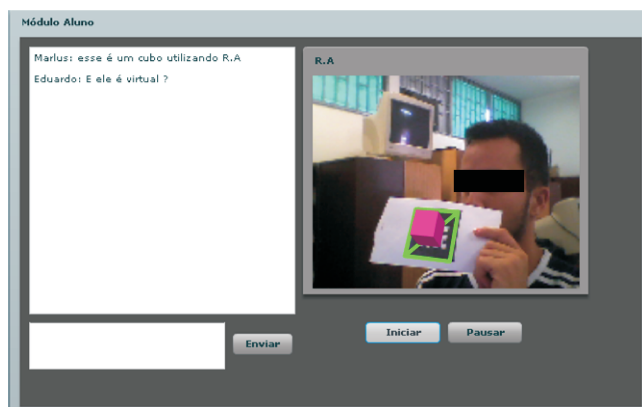


Figura 5. Interface do Módulo Aluno

A Figura 6 mostra que, o aluno pode interagir com o Módulo Aluno da seguinte forma: aceitando a transmissão das imagens enviadas pelo servidor, pausando em um determinado ponto, e voltar a receber as imagens. Também possui um *chat* para que possa interagir não somente com o professor, mas também com os demais alunos.

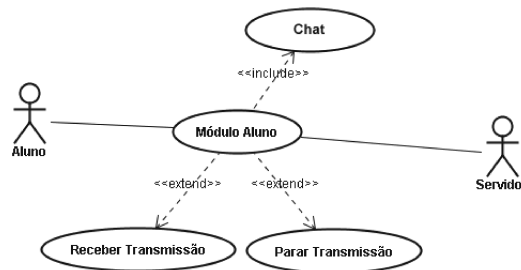


Figura 6. Caso de Uso Aluno

5 Conclusões e Trabalhos Futuros

Nos testes realizados foram obtidos resultados bastante interessantes, já que o acesso de 10 máquinas pela *web* foi possibilitado causando uma leve sobrecarga no sistema, com os clientes recebendo cerca de 30 quadros por segundo (FPS), elevando o processamento do servidor a 100%. Com 12 FPS, foram suportados 30 clientes sem impactos no desempenho e provocando uma média de processamento de 80% no servidor. Não foram realizados testes com uma quantidade acima de 30 clientes, no entanto, com o aprimoramento do projeto, pretende-se efetuar uma validação do sistema com turmas mais extensas em um ambiente de EaD.

Como trabalho futuro, pode-se desenvolver uma interface para *upload* de modelos e gerenciamento de marcadores, que possibilite ao Docente colocar seu próprio modelo nos formatos MD2, Collada ou 3ds não necessitando de recompilar o código do programa. Além disso, o desenvolvimento de um módulo que envie, ao invés da imagem, a matriz de transformação e outros parâmetros do objeto que esta sendo utilizado na apresentação, traria grande benefícios em relação à redução do tráfego de rede.

Referências

- [1] Adobe. Flash player penetration, 2009. Disponível em: http://www.adobe.com/products/player_census/flashplayer/. Acesso em: 13 agosto. 2009.
- [2] G. Avenue and M. View. *Java Remote Method Invocation Specification*. California 94043-1100 U.S.A., 1997.
- [3] R. T. Azuma. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, 21:34–47, 2001.
- [4] A. L. J. Cardoso. *Realidade Virtual: Uma Abordagem Prática*. SVR, 2004.
- [5] W. A. da Silva. Uma arquitetura para distribuição de ambientes virtuais de realidade aumentada. Master's thesis, Universidade Federal de Uberlândia, 2008.
- [6] M. W. de Souza Ribeiro. *Arquitetura para distribuição de ambientes virtuais multidisciplinares*. PhD thesis, Universidade Federal de Uberlândia, 2006.

- [7] C. Filhoais and J. Trindade. Física no computador: O computador como uma ferramenta no ensino e na aprendizagem das ciências físicas. *Revista Brasileira de Ensino de Física*, 25, 2003.
- [8] FluorineFx. Flex, flash remoting flex data services real-time messaging, 2009. Disponível em: <<http://www.fluorinefx.com/>>. Acesso em: 08 agosto. 2009.
- [9] W. Grosso. *JAVA RMI*. O'Reilly, 2001.
- [10] Microsoft. Internet information services, 2009. Disponível em: <<http://www.iis.net/>>. Acesso em: 08 agosto. 2009.
- [11] L. F. Silva. *Ambientes distribuídos em Realidade Virtual como suporte a Aprendizagem Cooperativa para a Resolução de Problemas*. PhD thesis, Universidade Federal de Uberlândia, 2009.
- [12] A. Tanenbaum and M. Steen. *Sistemas Distribuídos: Princípios e Paradigmas*, 2008.