

Novas Tecnologias para o Desenvolvimento de Conteúdo 3D para Web

Marcilio O.O. Lemos

LabTEVE
Universidade Federal da Paraíba (UFPB)
João Pessoa/PB - Brasil
mool@di.ufpb.br

Liliane S. Machado

LabTEVE / Departamento de Informática
Universidade Federal da Paraíba (UFPB)
João Pessoa/PB - Brasil
liliane@di.ufpb.br

Abstract— As tecnologias para o desenvolvimento de ambientes 3D para web evoluíram nos últimos anos para permitir a criação e disponibilização de cenários mais interativos e robustos graficamente. Este trabalho analisa e compara três tecnologias para o desenvolvimento de ambientes virtuais e conteúdo 3D para web, expondo as características de cada ferramenta e realizando um comparativo entre elas quanto à integração com a web e uso de recursos de realidade virtual, dentre outros. Também é apresentado um estudo de caso realizado com o desenvolvimento de um ambiente virtual com a finalidade de testar e comparar a qualidade gráfica e o suporte a características de realidade virtual oferecidos para cada uma destas ferramentas.

Keywords— Realidade virtual, luva de dados, modelos compostos, desenvolvimento de aplicações.

I. INTRODUÇÃO

No contexto da Realidade Virtual podemos definir ambientes virtuais como uma simulação gráfica em tempo real onde o usuário pode interagir com os objetos deste ambiente dentro de um espaço de referência [1]. Estes ambientes virtuais estão cada vez mais presentes no contexto da web, que representa um meio de possibilitar que cada vez mais pessoas tenham acesso a estes ambientes de uma forma simples e eficiente, havendo apenas a necessidade que o usuário tenha acesso a internet.

As linguagens de modelagem de ambientes virtuais já possibilitavam que os ambientes virtuais para Web fossem executados em navegadores (específicos ou não) mediante ao uso de *plug-ins*. Apesar de todas as vantagens que estas linguagens ofereciam havia a necessidade de que os ambientes virtuais pudessem ser executados em qualquer navegador sem a necessidade de *plug-in* ou interfaces de *plug-ins*. Atualmente, ferramentas como a WebGL e o X3DOM já são suportadas pela maioria dos navegadores padrão e permitem que o ambiente virtual se torne um elemento do contexto html.

Neste trabalho são analisadas três ferramentas de desenvolvimento de ambientes virtuais e conteúdo 3D para

a web, onde cada uma delas terá suas características descritas e comparadas com as demais.

II. METODOLOGIA

Dentre as tecnologias para o desenvolvimento de conteúdo 3d para web, três se destacam: X3D, WebGL e a X3DOM.

Além do fato de serem as tecnologias mais recentes, diversos fatores motivaram o estudo das mesmas por parte deste trabalho, sendo alguns pontos chave:

- A especificação do html5 definiu o X3D como padrão para a cenas 3d declarativas na web [2].
- A WebGL possui suporte por parte dos principais navegadores, permitindo assim que a cena 3d possa ser executada na web sem a necessidade de um *plug-in*
- O X3DOM realiza a integração do html5 com o padrão X3D e opera utilizando a WebGL, o que também o torna independente de interface externa.

A. X3D

O X3D é um padrão 3D para web [3] que descreve formas e comportamentos de ambientes virtuais. Surgido como uma extensão da linguagem VRML, o X3D utiliza a codificação XML, tornando sua arquitetura modularizada, e incorpora todas as vantagens da VRML, além de novas características adicionais. Na estrutura do X3D o ambiente virtual é representado como um grafo de cena e pode ser executado em qualquer navegador específico que ofereça suporte ao seu formato.

O desenvolvedor que utiliza o X3D se beneficia de uma rica API [4], com diversos nós organizados em componentes distribuídos em perfis. Ainda é possível criar seus próprios nós fazendo o uso de nós “prototipados”. O X3D também oferece suporte a outras linguagens e *scripts* para desenvolver o comportamento da cena utilizando uma interface conhecida como SAI (*Scene Access Interface*) [5]. A SAI permite alterações no grafo de cena em tempo de execução, sendo o acesso feito internamente pelo X3D com auxílio de um *script*.

B. WebGL

A WebGL é uma API multiplataforma baseada no OpenGL ES 2.0, para criar gráficos 3D que podem ser executados em um navegador web sem a necessidade de *plug-in* [6]. Esta API faz a ligação do JavaScript com funções do OpenGL ES, tornando assim possível a utilização de aceleração de *hardware* 3D em objetos ou cenas 3D contidas em páginas da web. Ela também utiliza o elemento Canvas do html5 para desenhar o objeto 3D na página, que é acessado por meio de Interfaces DOM (*Document Object Model*), e o gerenciamento automático de memória é fornecida como parte da linguagem JavaScript [6].

Com a WebGL é possível desenvolver conteúdo 3D complexo uma vez que esta é uma API de baixo nível. Diversas bibliotecas vêm sendo desenvolvidas para a WebGL com o objetivo de tornar o desenvolvimento das aplicações 3D mais simples e de mais alto nível, podendo ser citadas como exemplo as bibliotecas WebGLU, SpiderGL e a CD3L.

C. X3DOM

X3DOM é um *framework* de código aberto desenvolvido para preencher a corrente especificação do html5 para conteúdo 3D declarativo [7]. A W3C definiu o X3D como linguagem padrão para o desenvolvimento 3D para web em páginas html, logo o X3DOM integra uma cena X3D com o html5, onde esta cena é representada como um elemento da árvore DOM de um documento html. Nenhuma interface de *plug-in* ou *plug-ins* específico (como o SAI para o X3D) é necessário, pois a cena X3D é renderizada a partir da WebGL.

A arquitetura deste *framework* cria uma representação da cena X3D e sincroniza esta representação com a árvore DOM, para que assim a cena possa ser renderizada. Qualquer atualização na árvore DOM deve ser refletida na árvore X3D, assim como qualquer mudança na árvore X3D deve alterar a árvore DOM [8]. *Stream* de mídia permitem que elementos de mídia externos possam ser referenciados por elementos da árvore X3D.

Inicialmente foi criado um perfil do X3D específico que define todos os nós e elementos para integração com o html5, onde alguns nós do X3D foram excluídos e novos nós foram implementados com o objetivo de generalização, como por exemplo, o nó *Texture*. Atualmente já existe um perfil mais completo que possui quase toda especificação do X3D e já é possível adicionar novos tipos de nós ao contexto do X3DOM.

Com a definição de um *Namespace* [9] para representar o X3D no DOM, elementos do mesmo podem ser acessados e manipulados por métodos DOM e eventos podem ser aplicados a esses elementos, e o X3DOM adiciona o uso do CSS para desenvolver

animações, realizar transformações 3D e varias outras formas de eventos para interação com o usuário e sistema de monitoramento.

III. ESTUDO DE CASO: UM AMBIENTE INTERATIVO

Como estudo de caso, foi desenvolvido um ambiente virtual 3d interativo com o objetivo de analisar a qualidade gráfica oferecida por cada uma das ferramentas estudadas, bem como o suporte as características que compõe um ambiente virtual pelos mesmos. Nestes ambientes foram representados uma sala com alguns objetos. Este mesmo ambiente foi desenvolvido utilizando cada uma das ferramentas descritas neste trabalho. No ambiente foram implementadas características como iluminação, textura, navegação pelo ambiente, sobreposição de objetos. Também temos uma simples animação, caracterizada por um globo girando, e uma interação com usuário. Foi utilizado o mesmo tipo de textura e iluminação para cada um dos ambientes virtuais desenvolvidos e todos eles possuem as mesmas características quanto a forma e a modelagem.

Primeiramente o ambiente foi implementado utilizando o X3D. Devido a sua Diversificada API, questões como modelagem do ambiente, utilização de texturas e iluminação foram realizadas de forma simples e de alto nível. O ambiente foi testado utilizando o *browser* Xj3D.

Para o desenvolvimento do ambiente usando o X3DOM, não foi utilizado o perfil completo do X3DOM. Para testar a integração com o html5, foram utilizados métodos e eventos do Javascript para realizar as funcionalidades de alguns nós que não são suportados pelo perfil básico do X3DOM e que foram utilizados na concepção do ambiente utilizando o X3D. Neste contexto, o nó TouchNode [5] foi utilizado para finalizar a animação presente no ambiente virtual X3D, sendo que no ambiente virtual X3DOM foi utilizado o método onclick [10] do JavaScript para realizar a mesma função. Características presentes no *browser* Xj3D, como mudança de câmera, também foram implementadas na página onde se encontra o Ambiente Virtual do X3DOM, utilizando elementos do html5. O resultado final pode ser observado na Figura 1, na qual os ambientes desenvolvidos utilizando o X3D e o X3DOM estão representados.

Outro ambiente foi implementado usando a WebGL, utilizando as bibliotecas glMatrix [11] para trabalhar com matrizes e a WebGL-utils [12] para contornar problemas de não portabilidade entre o protótipo da função de redesenho usada pela WebGL. Devido ao fato de se ter utilizado apenas uma API de baixo nível, este ambiente foi o que apresentou maior grau de dificuldade para sua implementação e o a qualidade do ambiente está limitado pela capacidade de cada ferramenta.

O ambiente desenvolvido utilizando a WebGL foi o que mais se diferenciou dos demais devido à sua capacidade de

aceleração de *hardware* 3D, como pode ser observado na Figura 2.

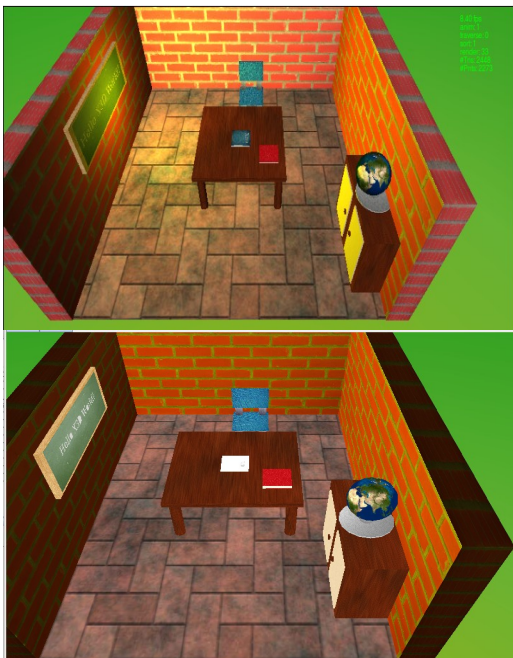


Figura 1. ambiente virtual utilizado como estudo de caso, desenvolvido utilizando o X3DOM e o X3D respectivamente.

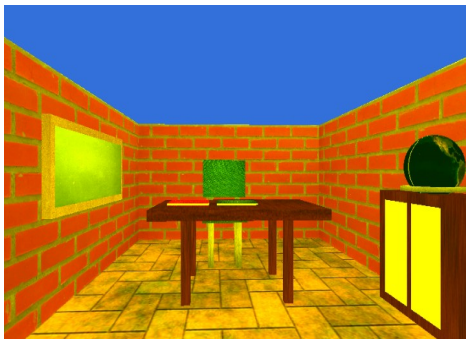


Figura 2. ambiente virtual desenvolvido utilizando a WebGL.

Este fato fica evidente se observarmos a intensidade da iluminação que está presente no ambiente criado usando a WebGL se comparada as demais, mesmo que os valores atribuídos a iluminação sejam os mesmos em cada ambiente.

Os resultados obtidos por este estudo de caso estão explícitos na Figura 3, onde uma escala de qualidade foi elaborada pelo desenvolvedor dos ambientes. Quesitos como taxa de quadros por segundo e tempo de desenvolvimento também foram considerados. Neste caso, a WebGL foi a que demandou maior tempo de desenvolvimento, particularmente devido à maior gama de

opções disponíveis ao programador. Por sua vez, o X3D apresentou a maior taxa de quadros por segundo.

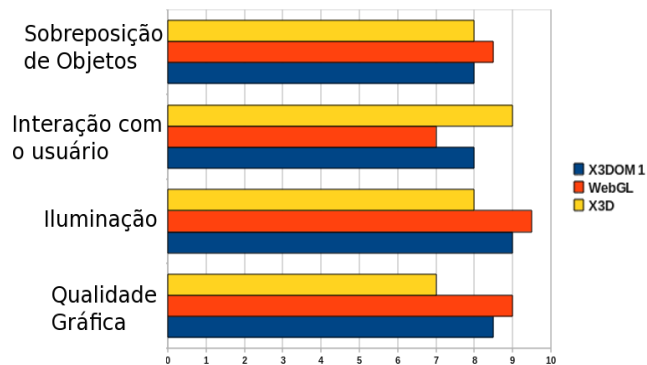


Figura 3. Resultados do estudo de caso utilizando ambientes virtuais desenvolvidos com diferentes tecnologias..

IV. RESULTADOS

As características, bem como as vantagens e desvantagens das tecnologias 3D para *web* aqui estudadas, foram analisadas de diferentes pontos considerados importantes para o desenvolvimento de conteúdo 3D interativo no contexto da *web*. Também foi explorado o suporte em navegadores para dispositivos móveis para estas tecnologias.

Um ponto importante a ser destacado é a capacidade de adicionar novos recursos para o desenvolvimento do ambiente virtual e não deixar o desenvolvedor limitado a especificação do meio de desenvolvimento utilizado. Neste quesito todas as três ferramentas obtiveram êxito, sendo que o X3D oferece esse recurso como parte de seu próprio contexto através de “nós prototipados” (*prototype node*) [5] enquanto que o X3DOM e a WebGL utilizam-se de recursos do html5 e da linguagem Javascript para este propósito.

Sob o aspecto da portabilidade, considerou-se importante que os Ambientes Virtuais pudessem ser acessados da *web* não apenas por um computador convencional de mesa, mas por qualquer dispositivo com acesso a Internet. Observa-se que já existem versões de navegadores X3D para dispositivos móveis e a maioria deles oferece suporte a WebGL [13]. O fato do X3DOM operar utilizando a WebGL faz com que navegadores que oferecem suporte à WebGL também ofereçam suporte a este *framework*.

Do ponto de vista de integração com a *web* o X3D apresenta a desvantagem no quesito dependência, já que o mesmo necessita de um navegador que seja especificamente desenvolvido para sua execução. Por outro lado, a WebGL e o X3DOM não apresentam esta dependência. Para eles, a cena 3D faz parte do contexto da própria página html, sendo o X3DOM como parte do elemento DOM e a WebGL

como parte da linguagem Javascript e ambos podem se comunicar com qualquer elemento da página. Entretanto, com relação à integração com outras linguagens, estas últimas duas necessitam de uma interface específica, com exceção da linguagem Javascript.

Foi comprovado através do estudo de caso que a WebGL se destaca das demais quanto à qualidade gráfica e desempenho. Utilizando recursos de baixo nível da OpenGL ES 2.0 ela permite manipular a renderização da cena 3D. Em conjunto com a WebCL, ligações do Javascript com o padrão OpenCL Kronos para computação paralela heterogênea permitirão em breve [14] ao desenvolvedor utilizar o processamento paralelo da GPU para melhorar a aceleração gráfica de sua aplicação. No X3DOM não há a mesma liberdade de explorar diretamente a qualidade gráfica da cena e há restrição das configurações preestabelecidas para que o X3DOM tenha sua cena renderizada utilizando a WebGL. Já no caso do X3D, este foi o que menos se destacou no quesito qualidade gráfica, mesmo utilizando a OpenGL.

O X3D se sobressaiu no quesito suporte à realidade aumentada, ou seja, permite a sobreposição de objetos virtuais no ambiente físico [15] e sua manipulação em tempo-real. O *framework* InstantReality [16] adiciona nós ao X3D que permitem desenvolver ambientes virtuais com características de realidade aumentada, oferecendo suporte para alguns dispositivos que permitem a interação entre o ambiente e o usuário. Dentro deste mesmo contexto de realidade aumentada pode-se utilizar dispositivos hápticos para proporcionar a sensação de toque, sendo que o X3D possui diversas APIs para criação de cenas hápticas. Por se tratar de tecnologias mais recentes, o X3DOM e a WebGL possuem pouco suporte à realidade aumentada e, geralmente, este suporte está limitado a um sistema operacional específico.

V. CONCLUSÕES

Neste artigo são apresentadas as principais e mais recentes tecnologias para o desenvolvimento de conteúdo 3D e ambientes virtuais na internet, onde suas características mais relevantes foram destacadas, bem como suas vantagens e desvantagens. Este trabalho poderá servir de apoio a desenvolvedores e pesquisadores com relação a escolha de qual ferramenta pode ser utilizada em seus trabalhos.

O ambiente utilizado como estudo de caso teve suas características moldadas de acordo com as capacidades de cada ferramenta quanto aos quesitos da qualidade gráfica, embora todos os elementos essenciais de um ambiente virtual tenham sido representados.

O X3D foi, por alguns anos, a principal forma de descrição de conteúdo 3D na web juntamente com seu antecessor o VRML. Porém, com o lançamento da

WebGL e do X3DOM, as possibilidades de integração dos ambientes expandiram-se uma vez que elas são mais amplas e robustas. Por esta razão, as mesmas vêm recebendo atenção tanto pela comunidade acadêmica quanto por grandes empresas como é o caso da Google em relação à WebGL.

Por serem relativamente recentes, a WebGL e o X3DOM ainda não possuem um suporte padrão e multiplataforma para realidade aumentada. Esta é certamente uma área de pesquisa promissora.

AGRADECIMENTOS

Este trabalho é apoiado pela universidade Federal da Paraíba e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico por meio de bolsa de estudo PIBIC.

REFERÊNCIAS

- [1] Moshell, J.M., & Hughes, C.E. 2002. *Virtual environments as a tool for academic learning*. Handbook of virtual environments, 893–910. Acessado em 30/04 de 2010. Online: <http://depts.washington.edu/edtech/moshellHughes.pdf>
- [2] W3C, 2012. *The Declarative 3d scenes in html5*. <http://dev.w3.org/html5/spec/Overview.html#declarative-3d-scenes>
- [3] Web3D Consortium, 2008. *X3D*. Online: <http://www.web3d.org/x3d>. Acessado em março de 2012.
- [4] Web3D Consortium, 2012. *X3D Specifications*. Online: <http://www.web3d.org/x3d/specifications/>. Acessado em março de 2012.
- [5] Web3D Consortium, 2012. *Scene access interface (SAI)*, iso/iec cd 19775-2 ed. 2:200x. Online: <http://www.web3d.org/x3d/specification/ISO-IEC-CD-19775-2.2-X3D-SceneAccessInterface/>. Acessado em março de 2012.
- [6] Kronos, 2012. *WebGL – OpenGL ES 2.0 for the web*. Online: <https://www.khronos.org/WebGL/>. Acessado em março de 2012.
- [7] X3DOM, 2010. *X3DOM*. Online: http://www.x3dom.org/?page_id=2. Acessado em março de 2012.
- [8] Behr, J., E Schler, P., Jung, Y., and Z Ollner, M. 2009. *X3DOM – a DOM-based HTML5/ X3D integration model*. In Proceedings Web3D '09, ACM Press, New York, USA, 127–135.
- [9] W3C, 2012. *Namespaces in xml*. W3C Consortium. Online: <http://www.w3.org/TR/REC-xml-names/>. Acessado em março de 2012.
- [10] W3C, 2009. *References to HTML elements from a script*. Online: <http://www.w3.org/TR/html4/interact/scripts.html>. Acessado em março de 2012.
- [11] Toji, 2012. *GL-Matrix*. Online: <https://github.com/toji/gl-matrix>. Acessado em março de 2012.
- [12] Google Inc, 2012. *WebGL-utils*. Online: <https://cvs.khronos.org/svn/repos/registry/trunk/public/WebGL/sdk/demos/common/WebGL-utils.js>. Acessado em março de 2012.
- [13] X3DOM, 2012. *X3DOM goes mobile*. Online: <http://www.x3dom.org/?p=825/>. Acessado em março de 2012.
- [14] Kronos, 2012. *WebCL - Heterogeneous parallel computing in HTML5 web browsers*. Online: <https://www.khronos.org/webcl/>. Acessado em março de 2012.
- [15] Kirner, C. and Siscoutto, R. *Realidade Virtual e Aumentada: Conceitos, Projeto e Aplicações*. Editora SBC – Sociedade Brasileira de Computação, Porto Alegre, 2007. Livro do pré-simpósio, IX Symposium on Virtual and Augmented Reality, Petrópolis – RJ, 2007.
- [16] Fraunhofer IGD and ZGDV, 2012. InstantReality. Online: <http://www.instantreality.org/story/what-is-it/>.