

Benefícios e aplicabilidade do WebGL na construção de ambientes virtuais tridimensionais por designers

Benefits and applicability of WebGL in the construction of three-dimensional virtual environments by designers

Bruno Spanevello Pergher

Doutorando em Design pela Universidade Federal de Santa Maria (UFSM) - bruno.pergher@gmail.com

José Luís Farinatti Aymone

Doutor em Engenharia Civil pela Universidade Federal do Rio Grande do Sul (UFRGS). Professor da Universidade Federal do Rio Grande do Sul (UFRGS) - aymone@ufrgs.br

Resumo

Este trabalho busca definir os benefícios da utilização da tecnologia WebGL, por designers, na construção de ambientes virtuais tridimensionais interativos e avaliar a possibilidade de inserção da mesma nos métodos de trabalhos destes profissionais. Para isso, foi realizada uma revisão bibliográfica com o intuito de esclarecer o contexto e especificidades do problema. Com as informações obtidas, definiram-se critérios e requisitos para avaliação de 26 ferramentas encontradas que trabalham de forma direta ou indireta com a criação de conteúdo em WebGL. Dentre elas, foi selecionado o conjunto de *softwares* (Blender + Blend4Web) que melhor cumpria os requisitos e, por consequência, melhor poderia solucionar a questão. Esse foi então testado em profundidade quando utilizado na criação de um ambiente virtual composto por objetos 3D, animações e interações. Analisando os resultados, desvendaram-se diversos benefícios trazidos pelo WebGL como a visualização em multiplataforma sem a necessidade de qualquer complemento (*plug-in*). Confirmou-se também a aplicabilidade dele nos projetos de Design Virtual que ganham em qualidade gráfica e interatividade, podendo ser distribuídos pela internet ou meios off-line.

Palavras-chave: Design virtual. Ambientes tridimensionais. WebGL.

Abstract

This work aims to define the benefits of the use of WebGL technology by Designers in the construction of interactive three-dimensional virtual environments and to evaluate the possibility of insertion of it into the work methods of these professionals. For this, a bibliographical review was carried out to clarify the context and specificities of the problem. With the information obtained, we defined criteria and requirements for the evaluation of 26 tools found that work directly or indirectly with the creation of content in WebGL. Among them, it was selected the software suite (Blender + Blend4Web) that best met the requirements and could better solve the question. This was then thoroughly tested when used in creating a virtual environment composed of 3D objects, animations and interactions. Analyzing the results, several benefits brought by WebGL as multiplatform visualization were unveiled without the need of any plug-in. It was also confirmed the applicability of it in the Virtual Design projects that gain in graphic quality and interactivity, being able to be distributed through the internet or offline means.

Keywords: Virtual design. Three-dimensional environments. WebGL.

Recebido em: 09/04/2017

Aceito em: 28/11/2017

1 INTRODUÇÃO

Os ambientes virtuais tridimensionais têm seu campo de aplicações cada vez mais estendido à medida que o avanço da tecnologia computacional aprimora as capacidades tecnológicas para criação e utilização deles. Mesmo tendo surgido décadas atrás, a Realidade Virtual (RV) é hoje concreta e viável, se tornando a grande aposta tecnológica para os próximos anos e faturando, segundo a empresa de auditoria, assessoria e consultoria empresarial Deloitte, aproximadamente 700 bilhões dólares em vendas de hardware e 300 bilhões em vendas de conteúdo para computadores, videogames e plataformas móveis em 2016 (LEE; STEWART, 2016). Antonio Rabello, em entrevista à Revista Exame (DINO, 2016), explica o que é necessário para manter este crescimento:

Sem conteúdo não há crescimento, porque não há o que ser consumido. Neste processo, os estúdios de produção imersiva têm um papel fundamental e ao mesmo tempo desafiador: criar em quantidade e com qualidade para atender a demanda, diz Antônio Rabello, diretor do VR Studios.

Para melhorar progressivamente a qualidade dos conteúdos de realidade virtual e garantir a contínua expansão deste mercado, são necessários conhecimentos de Design e sem dúvida os designers se apresentam como importantes profissionais no desenvolvimento de ambientes virtuais, em especial os tridimensionais. No entanto, esta é uma indústria multidisciplinar, destacando-se os que possuem um conhecimento mais abrangente, não só de Design, mas também das tecnologias de criação que hoje estão muito mais próximas dos programadores.

Algumas das principais tecnologias de descrição de cenas para *web* foram desenvolvidas até o ponto que usuários sem conhecimento algum de programação também pudessem gerar conteúdo. Isso aconteceu no final dos anos 90 e começo dos anos 2000 com o VRML¹ por exemplo, primeiro padrão de descrição de cenas 3D voltadas a *web*, que depois evoluiu para o x3d e foi essencial para a popularização e crescimento das pesquisas entorno do assunto. Hoje estamos diante do WebGL, um padrão de processamento de imagens e efeitos gráficos acelerados pela Unidade de Processamento Gráfico (GPU, sigla em inglês) de computadores e dispositivos móveis,

¹ O VRML ou *Virtual Reality Modeling Language* é definido por Marins; Haguenaer; Cunha (2007, p. 03), como uma linguagem textual que descreve objetos e suas inter-relações em um espaço virtual tridimensional. Não existindo procedimentos de lógica, apenas de representação.

permitindo que conteúdos tridimensionais sejam incorporados a páginas *web* (HTML) e executados em navegadores *web* compatíveis (ARORA, 2014, p. 35).

Muito mais poderoso que seus antecessores, o WebGL traz consigo um salto de qualidade nunca antes visto, possibilitando a distribuição de jogos, animações, simulações científicas, apresentações interativas e tantas outras aplicações interativas pela internet e para muitas plataformas. Hoje, todos os principais dispositivos móveis (*smartphones* e *tablets*) ou “de mesa” (computadores e videogames) e sistemas operacionais (Windows, Mac OS, Linux) já possuem a capacidade de exibir conteúdo 3D *online* e *off-line* por meio desta tecnologia.

A inclusão de designers nesse meio parece natural e necessária para alcançar o aprimoramento constante da qualidade dos produtos digitais. Resta apenas descobrir por quais caminhos o WebGL pode iniciar a sua aproximação com estes profissionais, proporcionando-lhes mais conhecimento e aperfeiçoamento de suas capacidades técnicas e inventivas.

Vislumbrando os possíveis benefícios proporcionados pela tecnologia WebGL aplicada à criação de ambientes interativos tridimensionais (3D) no Design, acredita-se que a criação de uma sistemática composta por ferramentas que compreendam o WebGL desde a modelagem de objetos 3D, concepção de cenários com animações e interação até a finalização do projeto, pode oferecer maior qualidade de apresentação e potencializar a abrangência dos trabalhos de designers.

Imagina-se que seja possível oferecer aos designers uma maneira mais simples de aproximação e utilização das funcionalidades do WebGL, ampliando suas capacidades em projetos e dando-lhes a oportunidade de contribuir com suas criações neste universo que promete evoluir muito nos próximos anos e se tornar um dos pilares da computação gráfica (LEVKOWITZ; KELLEHER, 2012).

São vários os possíveis enriquecimentos que a tecnologia WebGL pode trazer aos trabalhos de Design. Pode-se, por exemplo, elevar a qualidade de apresentação de projetos de design, com visualizações interativas inseridas em contextos em plena evolução como a Realidade Virtual. Ou ainda, disseminar um conteúdo com maior facilidade, devido a sua capacidade de execução multiplataforma, inclusive em dispositivos móveis (*smartphones* e *tablets*), sem a necessidade de complementos específicos e por meio de simples navegadores *web* (ZHANG; GRAČANIN, 2013).

A aproximação da tecnologia WebGL dos designers pode popularizá-la e ajudar na sustentação da própria tecnologia como padrão definitivo de descrição de cenas tridimensionais

para a *web*. Porque segundo Danchilla (2012), um dos pilares para alcançar este patamar é a adoção maciça da comunidade de desenvolvimento, hoje composta em sua maioria por programadores e por vezes carente de designers.

Os benefícios podem atingir ainda o campo da educação. Semelhante ao realizado pelo Prof. Dr. José Luís Farinatti Aymone na disciplina, ministrada por ele, de Design Virtual da pós-graduação em Design da UFRGS com relação à criação de cenários 3D com o VRML, futuramente os conhecimentos de desenvolvimento de ambientes virtuais utilizando WebGL poderão ser transmitidos a acadêmicos e interessados em Design Virtual, tendo como base uma sistemática especialmente desenvolvida para os profissionais da área. Por sua vez, estes aprendizes poderão dar sequência à pesquisa nesta área.

2 METODOLOGIA

Diante do anseio de tornar os designers beneficiários das atuais tecnologias de criação de ambientes virtuais interativos, realizou-se uma revisão bibliográfica de entendimento, objetivando traçar uma linha evolutiva das tecnologias destinadas ao desenvolvimento destes, sejam por designers ou programadores. Como retorno desta, obtiveram-se os principais tópicos a serem explorados a respeito das tecnologias de descrição de cenas virtuais tridimensionais, partindo do VRML, passando pelo X3D e chegando ao WebGL, que configura hoje o estado da arte no que diz respeito ao tema.

Identificou-se, porém, que apesar de oferecer aos designers vantagens significativas em comparação a seus antecessores, o promissor WebGL ainda está distante de designers que não possuem um conhecimento mínimo de programação, ainda que estejam familiarizados com *softwares* de modelagem 3D. Então, foram efetuados alguns ensaios livres de rigor metodológico, explorando-se ambientes virtuais dinâmicos, desenvolvidos a partir do VRML, do X3D e do WebGL, focando no reconhecimento de uma lacuna na área de pesquisa em questão. Em seguida realizou-se a caracterização do problema, com a clarificação de suas variáveis e elaboração da hipótese de solução através do método hipotético-dedutivo.

Tendo claras as condicionantes do problema, iniciou-se uma revisão bibliográfica de aprofundamento sobre as questões históricas, evolutivas e técnicas das tecnologias abordadas (VRML, X3D e WebGL). Em conjunto elaboraram-se discussões sobre assuntos elementares e de

compreensão necessária para a sequência da pesquisa, tais como: a tipificação e singularização da representação tridimensional virtual no *Design*, os campos da realidade virtual e aumentada, a qualificação dos ambientes tridimensionais virtuais em geral e a estruturação da programação orientada a objetos. Deste conteúdo observou-se os pontos mais importantes que foram traduzidos em critérios (**Erro! Fonte de referência não encontrada.**) para avaliação futura de *softwares* que trabalham com o universo de modelagem tridimensional virtual e exportação de conteúdo para WebGL.

Tabela 1 – Critérios utilizados para avaliação dos softwares de criação 3D virtual e WebGL.

Nº	Critérios
1	Tipo de licença (gratuita, paga, avaliação, educacional...)
2	Funcionamento <i>online</i> ou <i>off-line</i>
3	Sistemas operacionais suportados
4	Possibilidade de publicação direta em WebGL (html)
5	Exportação em código aberto ou fechado
6	Formatos de importação suportados
7	Formatos de exportação suportados
8	Funcionalidade (modelagem, animação, interação...)
9	Conhecimento prévio necessário
10	Periodicidade das Atualizações

Fonte: elaborado pelos autores.

Na sequência, deu-se início à fase de pesquisa exploratória prática com a coleta e reconhecimento de ferramentas relacionadas à criação de conteúdo com WebGL. Estas vão desde editores visuais de cenas 3D virtuais, passando por *plug-ins* e conversores, até hospedadores e visualizadores de conteúdo, preenchendo a necessidade de se conhecer os instrumentos de criação desde a modelagem do ambiente até a sua publicação, a fim de compreender as ligações entre o WebGL e o Design Virtual, bem como os benefícios para ambos. A avaliação das ferramentas elencadas se deu através do uso essencialmente prático das mesmas, buscando criar cenários compostos por modelos virtuais 3D e animações de movimentação simples, explorando as capacidades de inserção de interações mais simples como as de navegação ou mais complexas como a mudança intencional de estado dos objetos por meio de acionadores.

3 FUNDAMENTAÇÃO TEÓRICA

Para poder verificar o potencial de utilização e crescimento da tecnologia WebGL é necessário entender o que é a Realidade Virtual (RV) e a Realidade Aumentada (RA). A Virtual Reality Society ([201-]b) explica que realidade virtual é o termo utilizado para descrever um ambiente tridimensional, gerado por computador, que pode ser explorado e interagido por uma pessoa através de alguns dispositivos, como os óculos de realidade virtual (**Erro! Fonte de referência não encontrada.**). Essa pessoa torna-se parte deste mundo virtual ou está imerso nesse ambiente e lá é capaz de manipular objetos ou executar uma série de ações. Já a realidade aumentada é considerada por diversos autores como sendo uma tecnologia paralela a RV. Pan et al. (2006) diz que a RA, também chamada Realidade Mista, refere-se à incorporação de objetos gráficos virtuais em uma cena real tridimensional, ou, alternativamente, a inclusão de elementos do mundo real em um ambiente virtual. Enquanto a Realidade Virtual “mergulha” os sentidos em um mundo totalmente digital, a Realidade Aumentada projeta objetos digitais, como modelos 3D, vídeos, imagens e sons em nossa visão da realidade, como se realmente estivessem lá (**Erro! Fonte de referência não encontrada.**) (VIRTUAL REALITY SOCIETY, [201-]a).

Figura 1 – Oculus Rift, dispositivo importante para a popularização da RV nos últimos anos.



Fonte: Paget (2016).

Figura 2 – Exemplo de visualização de modelos virtuais sobrepostos a um ambiente físico a partir da Realidade Aumentada.



Fonte: Guéno (2014).

A partir destas definições e do sabido crescimento vertiginoso do mercado de Realidade Virtual, entende-se que os próximos conteúdos digitais deverão estar inseridos neste meio. Surge então uma nova necessidade de desenvolver ambientes tridimensionais virtuais que sejam possíveis de serem reproduzidos nas atuais e futuras plataformas de realidade virtual, tanto em computadores pessoais como em dispositivos móveis. Esse é o primeiro ponto positivo trazido pelo WebGL, que é capaz executar criações imersivas em diversos dispositivos, se aproveitando e se adaptando ao *hardware* de cada um. Além disso, as principais formas de exibição e interação de conteúdos para Realidade Virtual já são suportadas pela tecnologia.

Constituindo parte fundamental da Realidade Virtual, os ambientes tridimensionais virtuais possuem características peculiares que influenciam cada vez mais na forma com que são desenvolvidos. São definidos por Cunha e Mainente (2011) como cenários tridimensionais, gráficos e interativos gerados por computador, com o objetivo de possibilitar uma representação cibernética similar ao mundo físico, onde o usuário pode mover-se através dele e com recursos que propiciam uma sensação de profundidade e imersão.

Em vista disto, um ambiente virtual tridimensional (AV3D) possui ao menos quatro características essenciais que o identificam, como a evidente representação em três dimensões; a interatividade, no mínimo em nível de navegação; a imersão, em mundos que imitam a realidade ou cenários inteiramente inventados e a inserção em um espaço também tridimensional. Tal

espaço, que também é definido pelas dimensões altura, largura e profundidade, é aquele onde estão inseridos os objetos, animações e interações que em conjunto caracterizam um ambiente 3D por completo.

Então, fica claro que a capacidade de trabalhar com objetos tridimensionais, animações e interações é requisito obrigatório das ferramentas a serem utilizadas para a criação de AV3D contemporâneos. E mais uma vez o WebGL trabalha bem com animações simples e complexas, interações variadas e modelos tridimensionais com poucos ou muitos polígonos, já que seu processamento depende em sua maior parte de um melhor aproveitamento das capacidades de processamento do *hardware* dos dispositivos. Essas características combinadas possibilitam que diversas aplicações que antes eram desenvolvidas separadamente para cada plataforma (Computador, *Smartphone*, Videogame...), agora sejam executadas em navegadores *web* de maneira *online* ou *off-line* sem a necessidade de instalação no aparelho e sem a utilização de qualquer complemento especial (*plug-in*).

Esta tecnologia se mostra promissora porque resolve algumas das principais limitações das anteriores. Além de não serem mais necessários complementos para a visualização dos conteúdos, ela fornece funcionalidades de acesso direto ao *hardware* gráfico (GPU) dos dispositivos e as aplicações são executadas nativamente nele, permitindo uma aceleração gráfica mais eficiente, utilizando melhor a capacidade de processamento de computadores, *smartphones*, *tablets* e tantos outros (CONGOTE et al., 2011; LOESCH; CHRISTEN; NEIBIKER, 2012). Com isso, houve um salto de qualidade gráfica e a possibilidade de visualização de efeitos de física das cenas tridimensionais. Cenas estas, que podem agora ser visualizadas com melhor qualidade e em uma gama muito maior de aparelhos através de navegadores *web*, característica enfatizada por Russell et al. (2015, p. 03):

WebGL expõe o poder do processador gráfico que já está em todos os dispositivos, diretamente para a *web*, de maneira segura. Isso permite que aplicativos 3D sofisticados, poderosos e portáteis possam ser entregues ao usuário final, diretamente no navegador.

Os possíveis usos do WebGL são inúmeros e crescentes à medida em que se aumenta o poder de processamento dos dispositivos, em especial os móveis (*smartphones*, *tablets* e tantos outros), possibilitando a execução de aplicações com maiores exigências computacionais, diretamente de um navegador *web* e adaptável ao *hardware*. Já se encontram, para execução

multiplataforma e sem a necessidade de *plug-ins* através do WebGL, jogos com gráficos 3D detalhados e física realística, demonstrações virtuais de produtos (**Erro! Fonte de referência não encontrada.**) com animações e interações imersivas, completos *softwares online* de modelagem tridimensional e também editores próprios de ambientes virtuais em WebGL. Todas aplicações inimagináveis até poucos anos.

Figura 3 – Demonstração interativa de modelo virtual 3D realístico de um carro recém lançado.



Fonte: Little Workshop (2015).

No entanto, ainda há um aspecto dos seus antecessores do qual o WebGL ainda não se apropriou por completo, a facilidade de geração de conteúdo. Enquanto o VRML se fortaleceu no início dos anos 2000 sendo um padrão fácil de ser trabalhado, onde pessoas com conhecimento básico de modelagem 3D já podiam exportar suas criações para o formato, o WebGL ainda está muito atrelado ao conhecimento de linguagens de programação escrita.

Apesar de o WebGL se comunicar diretamente com o *hardware* dos eletrônicos, o desenvolvimento das aplicações ocorre em nível mais alto², com o auxílio de bibliotecas de programação ou *Library*, em inglês. Elas nada mais são que conjuntos de funções e definições pré-escritas para um propósito definido e prontas para serem utilizadas em um nível mais elevado da

² O processo abstração exclui detalhes indesejados ou insignificantes de um problema a ser resolvido (AGUILAR, 2008, p. 648). Uma linguagem de programação de alto nível é como se chama, na Ciência da Computação, uma linguagem com um nível de abstração relativamente elevado, longe do código mais “bruto” de máquina (mais baixo nível), entendido somente por técnicos da área, e mais próximo à linguagem humana. Muitos dos termos no VRML por exemplo, são semelhantes aos utilizados para descrever imagens e objetos no mundo físico, facilitando a legibilidade e compreensão do código (COELHO, 2011).

programação por aqueles que não tem motivos para entender a linguagem “bruta” da máquina simplesmente para criar um aplicativo (ANDRADE, 2015; JÁCOME, [2010]). As bibliotecas são fundamentais porque reduzem a complexidade da programação, subtraindo e resumindo detalhes, com vistas de aumentar a produtividade, proporcionando uma maneira mais fácil e compreensível de se desenvolver aplicações, no caso deste trabalho ambientes virtuais tridimensionais (MINISTÉRIO DA CULTURA, 2006).

Atualmente a criação de ambientes virtuais WebGL utilizando bibliotecas de programação é a mais usual. No entanto, ainda que facilite a vida de programadores, a maior parte dos designers não possuem sequer conhecimento de alguma linguagem escrita de programação. Felizmente a especificação de dados e instruções a serem transmitidas a um computador não precisa necessariamente ser realizada por escrito. Existem hoje algumas linguagens compostas por ambientes de desenvolvimento e sintaxe majoritariamente visuais. De acordo com Santos (2013), as linguagens deste tipo procuram em geral facilitar a resolução de problemas de uma forma mais intuitiva, atrativa e natural, voltada especialmente para a autoaprendizagem. Tal consideração vai ao encontro de iniciativas como o Scratch (**Erro! Fonte de referência não encontrada.**), focado na autoaprendizagem da programação computacional por crianças a partir da montagem de blocos de comando.

Figura 4 – Parte de um *software* construído por meio dos blocos coloridos da ferramenta *online* Scratch.



Fonte: Fupicat (2016).

Os benefícios das linguagens visuais de programação vão ao encontro dos requisitos (**Erro! Fonte de referência não encontrada.**) para a aproximação de designers e WebGL, sendo alternativas no processo que visa integrar as ferramentas de modelagem tridimensionais já

utilizadas por estes profissionais e a inserção de animações e interações às cenas destinadas a *web* ou meios *off-line*.

Tabela 2 – Requisitos obrigatórios e desejáveis para a seleção dos softwares aptos a trabalhar com o WebGL na criação de ambientes virtuais tridimensionais no Design.

Nº	Requisito	Condição
1	Possibilitar a criação e edição de conteúdo sem a necessidade de conexão com a internet	Desejável
2	Funcionar nos principais sistemas operacionais do mercado (Windows, OS X e Linux)	Obrigatório
3	Permitir publicação em WebGL de maneira simplificada	Obrigatório
4	Permitir a exportação do conteúdo diretamente em um HTML executável	Desejável
5	Possibilitar a importação dos mais diversos formatos de arquivo	Obrigatório
6	Não exigir conhecimento prévio de programação	Obrigatório
7	Possuir suporte ativo e atualizações frequentes	Desejável
8	Possuir versão gratuita ou educacional completa	Desejável

Fonte: elaborado pelos autores.

4 DESENVOLVIMENTO

Uma das chaves para que esta mistura de tecnologias funcione é a existência de *softwares* que trabalham de maneira muito semelhante aos já utilizados para modelagem tridimensional no design, porém possuem a funcionalidade de acréscimo de interações e animações aliadas à exportação deste conteúdo para esta Interface de Programação de Aplicativos (em inglês, *Application Programming Interface* - API) denominada WebGL.

Neste trabalho foram levantadas 26 ferramentas (**Erro! Fonte de referência não encontrada.**) entre editores de cena, modeladores 3D, exportadores e visualizadores, que trabalham de forma direta ou indireta com a criação de conteúdo em WebGL. Todas elas passaram por um teste exploratório, com base nos critérios estipulados anteriormente, para identificação de suas funcionalidades e capacidades no que diz respeito ao desenvolvimento e exportação de cenários tridimensionais animados e interativos com o WebGL. O teste consistiu na inserção, edição, montagem e exportação de uma cena diretamente em um arquivo (html) executável, com animações e interações, se possível. As ferramentas foram então elencadas segundo o número de requisitos não cumpridos por elas, para futura seleção da mais adequada a pesquisa.

Tabela 3 – Ferramentas levantadas para análise de funcionalidades relacionadas a criação de conteúdo em WebGL.

Ferramenta	Funcionamento	Requisitos (Tabela 2) <u>NÃO</u> Cumpridos pela Ferramenta
Blender + Blend4Web	<i>Off-line</i>	-
Autodesk Maya 2016 + Inka3D	<i>Off-line</i>	n ^o 6
Unreal Engine	<i>Off-line</i>	n ^o 6
Unity + Assets	<i>Off-line</i>	n ^o 6
Clara.io	<i>Online</i>	n ^o 1, 4
Three.js	<i>Online e Off-line</i>	n ^o 5, 6
WebGLStudio.js	<i>Online e Off-line</i>	n ^o 5, 6
PlayCanvas	<i>Online</i>	n ^o 1, 4, 5
Blender + Sketchfab Exporter	<i>Online</i>	n ^o 1, 4, 6
Atomic Game Engine	<i>Off-line</i>	n ^o 2, 6, 7
Goo Create	<i>Online e Off-line</i>	n ^o 4, 5, 6
Autodesk Maya 2016 + WebGL Exporter For Maya	<i>Off-line</i>	n ^o 6, 7, 8
3ds Max + Sketchfab Exporter	<i>Online</i>	n ^o 1, 2, 4, 6
Vizor.io	<i>Online</i>	n ^o 1, 4, 5, 6
SculptGL	<i>Online</i>	n ^o 1, 4, 5, 6
3ds Max 2017 + WebGL Exporter for Max	<i>Off-line</i>	n ^o 2, 4, 6, 8
Rhino + Iris	<i>Off-line</i>	n ^o 2, 6, 7, 8
AutoCAD 2017 + WebGL Exporter For Autocad	<i>Off-line</i>	n ^o 2, 6, 7, 8
CopperCube	<i>Off-line</i>	n ^o 2, 6, 7, 8
FinalMesh	<i>Off-line</i>	n ^o 2, 6, 7, 8
Blender + Babylon Exporter	<i>Off-line</i>	n ^o 3, 4, 6, 7
3ds Max 2017 + Three.js Exporter	<i>Off-line</i>	n ^o 2, 3, 4, 6, 7
3ds Max 2017 + Babylon Exporter	<i>Off-line</i>	n ^o 2, 3, 4, 6, 7
Marmoset Toolbag	<i>Off-line</i>	n ^o 2, 4, 6, 7, 8
Autodesk Stingray 2017	<i>Off-line</i>	n ^o 2, 5, 4, 6, 7
Cl3ver	<i>Online</i>	n ^o 1, 4, 5, 6, 7, 8

Fonte: elaborado pelos autores.

Apesar do extenso número de *softwares* avaliados, a maior parte foi classificada como inadequado por oferecerem limitações que forçaram o abandono precoce da análise. A deficiência

dos *softwares* na inserção de animações e interações ou o grande nível de conhecimentos prévios de programação exigidos (Requisito nº6) para a criação delas fez com que o conteúdo gerado não se enquadrasse nos requisitos básicos obrigatórios (**Erro! Fonte de referência não encontrada.**) dos ambientes virtuais 3D e nas habilidades comuns dos designers. Ainda, muitos não puderam ser avaliados por completo, pois as versões de avaliação dos programas não davam acesso a funcionalidades necessárias, disponíveis somente nas versões pagas. Estes também contrapõem o requisito desejável (nº 8) de se utilizar *softwares* gratuitos ou com versão educacional completa especialmente no meio acadêmico para o ensino de estudantes de Design. Portanto, entre as ferramentas restantes, somente o *software* Blender em conjunto com o seu complemento Blend4Web foram capazes de cumprir satisfatoriamente todos os requisitos obrigatórios e os desejáveis, sendo os resultados obtidos com eles apresentados e discutidos a seguir.

5 RESULTADOS

Ao passo que os *softwares* Blender e Blend4Web em conjunto demonstraram potencial em um primeiro teste exploratório, um novo experimento mais aprofundado foi realizado para verificar em detalhes as funcionalidades oferecidas e que possibilitariam a difusão do WebGL como tecnologia útil e aplicável aos projetos de Design Virtual e ao ensino.

Figura 5 – Parte de ambiente demonstrativo evidenciando o modelo de um carro com a paleta de possíveis cores para ele e à direita, um espaço de estar.



Fonte: elaborado pelos autores.

O experimento compreendeu a criação de um cenário (**Erro! Fonte de referência não**

encontrada., Erro! Fonte de referência não encontrada. e Erro! Fonte de referência não encontrada.) composto por variados objetos 3D com animações e interações desenvolvidas a partir da linguagem de blocos embutida nos programas. Objetivou-se explorar e demonstrar a potencialidade dos principais recursos oferecidos, bem como analisar o desempenho na execução do arquivo final (HTML) em computadores e *smartphones* através dos navegadores *web* mais utilizados no mundo - Google Chrome (60,5%), Mozilla Firefox (15,6%), Microsoft Edge/Internet Explorer³ (15,5%)-, conforme pesquisa realizada em Abril de 2016 pela empresa StatCounter (SIMPSON, 2016).

Figura 6 – Espaço de estar com interação nos objetos sobre a mesa e palavras na parede que acionam a troca do tipo de câmera.



Fonte: elaborado pelos autores.

Figura 7 – Ao fundo, um helicóptero de brinquedo que voa pelo espaço.

³ O navegador *web* Microsoft Edge foi lançado recentemente como uma atualização do antigo Internet Explorer, também da empresa Microsoft. Por isso, somente a versão mais atual (Edge) é utilizada para comparação com os demais *softwares*.



Fonte: elaborado pelos autores.

De modo geral foi possível verificar que o complemento Blend4Web consegue trabalhar com a maior parte dos recursos do Blender relacionados à modelagem 3D, aplicação e edição de materiais e texturas, animações lineares, efeitos de física e partículas, iluminação, câmeras e outras configurações do espaço tridimensional, exportando com sucesso o conteúdo criado, diretamente em um arquivo HTML único e executável. Essa exportação bem-sucedida, na maioria dos casos, se deve ao fato de que, quando ativado, o Blend4Web retira da interface do Blender todos os recursos com os quais ele ainda não consegue trabalhar, restando para o usuário apenas aqueles suportados.

Além disso o Blend4Web adiciona ao sistema de programação visual, já incluso no Blender, novos blocos com funções que não necessitam conhecimento prévio de programação para o seu entendimento. A partir da relação dos blocos é possível criar inúmeras interações, entre elas, movimentação e visualização da cena; transformação de objetos, luzes e câmeras quanto a suas dimensões, posição, material, visibilidade e outros; acionamento de animações e sons; redirecionamento para páginas da *web* ou outros ambientes, etc. Ainda existe a possibilidade de inserção de interações mais complexas ou relacionadas a arquivos externos por meio de arquivos com códigos javascript, porém devido a este estudo ser voltado ao processo de trabalho de designers esse ponto não foi explorado.

É importante ressaltar que dentre os 26 *softwares* analisados, nenhum outro apresentou um conjunto de recursos tão completo e que aliasse modelagem, animação e interação de forma

familiar a designers que já trabalham com *softwares* de Desenho Assistido por Computador (CAD, sigla em inglês).

Ressaltam-se os recursos mais relevantes do conjunto Blender + Blend4Web relacionados à modelagem, animação e interação, observados durante as experimentações. Primeiro, o *software* Blender por si só possui uma interface muito semelhante aos demais programas de modelagem 3D do mercado, sendo familiar aos profissionais que já utilizam estes. Por sua vez, o complemento Blend4Web se utiliza desta mesma interface, se aproveitando da maior parte das ferramentas do Blender para fornecer os recursos necessários para a exportação dos ambientes utilizando a tecnologia WebGL. Dessa forma, praticamente toda a parte referente à modelagem tridimensional, configuração de luzes e câmeras, tratamento de materiais e mapeamento de texturas é suportada sem limitações significativas.

O conjunto de *softwares* também trabalha satisfatoriamente com uma ampla gama de tipos animação, como as que compreendem as transformações de objetos como um todo (dimensão, posição, material, visibilidade...), as animações de esqueletos que permitem a movimentação de personagens, as que trabalham com o áudio da cena, que pode ser tanto reproduzido no ambiente como um todo ou com emissão concentrada em um ponto, as que simulam os efeitos ao ar livre (água, vento, atmosfera...) e aquelas que retratam alguns efeitos de partículas como fumaça, fogo e respingos d'água.

Finalmente, as interações podem ser construídas na tela *Node Editor*, onde se encontra a linguagem de blocos do Blend4Web. Entre os blocos básicos disponíveis estão os que controlam sons; visibilidade, transformações e materiais de objetos; os tipos e movimentações de câmeras; as animações presentes na cena; o realce de objetos por meio de contornos e as ações executadas ao clique do mouse ou outro dispositivo suportado. Existem alguns outros blocos que possibilitam também guardar valores em variáveis, realizar operações matemáticas e verificar condições de igualdade. Estes últimos viabilizam a criação de interações com maior grau de complexidade sem que isso represente um aumento na dificuldade de utilização da presente linguagem.

Em paralelo a tudo isso, o conjunto Blender + Blend4Web proporciona diversos recursos nativos interessantes e coerentes com o caminho que segue a tecnologia digital atualmente. Um deles é a navegação e movimentação por meio de mouse, teclado, controles de videogames e telas de toque. Em aparelhos com giroscópio (*smartphones* e *tablets*) também é possível acessar a

visualização em tela dividida, própria para equipamentos como os óculos de realidade virtual, preparando os projetos para a nova era da RV. Outros recursos nativos são:

- **Fast Preview:** Uma forma de visualização rápida do projeto no navegador web, antes de se exportar o arquivo HTML final. Junto a ela são disponibilizados recursos para a identificação de erros e ajustes finos em tempo real nas propriedades da cena 3D. Ele proporciona maior controle no processo de desenvolvimento de ambientes e aplicações virtual tridimensionais.
- **Otimizações:** Verifica-se o suporte a otimizações comuns em tecnologias de representação anteriores como a suavização de geometrias e a cópia de elementos que compartilham informações de atributos entre si, tornando a exigência de processamento computacional menor para a execução do conteúdo e aumentando a taxa de quadros por segundo⁴ (FPS, sigla em inglês) no momento de sua reprodução.
- **Perfis de Qualidade Gráfica:** Para todos os projetos exportados diretamente em HTML são criados três perfis de qualidade gráfica (Baixo, Alto e Ultra) com diferentes níveis de aprimoramento de texturas, iluminação e outros. Isso amplia as chances de o ambiente ser executado com fluidez satisfatória em dispositivos de menor capacidade de processamento.

Para finalizar os testes, verificou-se a fluidez de reprodução do ambiente criado nos três navegadores mais utilizados e já citados, em suas versões mais atuais. Para isso, registrou-se as taxas mínimas e máximas de quadros por segundo (FPS) no momento da execução, obtida por meio do recurso *Fast Preview* do próprio Blend4Web (**Erro! Fonte de referência não encontrada.**). Todas as análises foram realizadas em um mesmo computador e com o mesmo perfil de qualidade gráfica (Alta) para não haver possíveis influências externas. Ressalta-se ainda que o valor máximo possível de ser atingido em qualquer navegador é 60 FPS, devido a uma trava do sistema.

Tabela 4 – Taxas mínimas e máximas de quadros por segundo (FPS) durante a execução do ambiente WebGL nos três principais navegadores. Quanto maior, melhor.

⁴ “Quadros por segundo (FPS) é uma medida de como o movimento de vídeo é exibido. O termo se aplica igualmente ao vídeo de filme e vídeo digital. Cada quadro é uma imagem estática” (MICROSOFT, [201-]). Quanto maior o número de quadros por segundo, maior a percepção visual de fluidez do conteúdo reproduzido.

	Google Chrome v.55	Mozilla Firefox v.50	Microsoft Edge v.38
Mínimo	58 FPS	55 FPS	57 FPS
Máximo	60 FPS	60 FPS	60 FPS

Fonte: elaborado pelos autores.

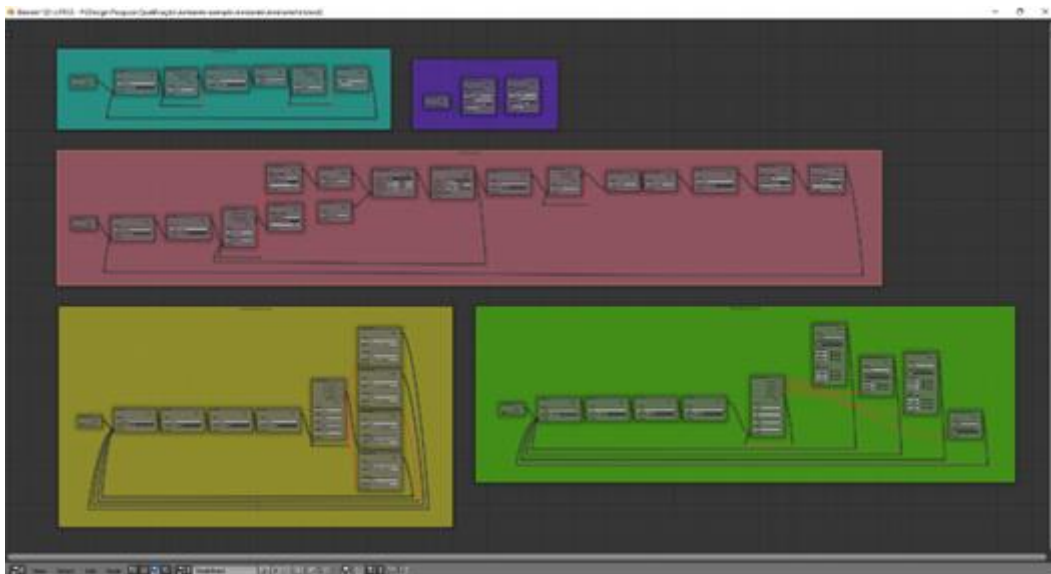
A partir da análise destes registros percebe-se que não há diferenças significativas quanto à fluidez de visualização do ambiente nos três navegadores, também não sendo possível diferenciar visualmente as pequenas quedas na taxa. Pode-se dizer que não há superioridade expressiva no uso de um destes navegadores para a execução de conteúdos gerados pelo conjunto Blender + Blend4Web, confirmando a possibilidade de uma distribuição mais abrangente dos mesmos.

5.1 VERIFICAÇÃO DOS REQUISITOS

Após a discussão dos resultados averiguou-se então o cumprimento ou não dos requisitos obrigatórios e desejáveis para os *softwares* aptos a trabalhar com o WebGL na criação de ambientes virtuais tridimensionais no Design, conforme a **Erro! Fonte de referência não encontrada..** Primeiramente, a associação destes dois *softwares* não exige a conexão direta com a internet para o seu funcionamento (requisito nº 1), permitindo que uma vez instalados no computador, se crie, importe ou edite modelos 3D e cenas completas. Também possuem versões para Windows, Mac OS e Linux (requisito nº 2), abrangendo os principais sistemas operacionais do mercado.

Proporcionam a publicação facilitada do conteúdo diretamente em um arquivo HTML executável nos principais navegadores *web* de computadores pessoais, *videogames*, *smartphones* e *tablets* (requisitos nº 3 e 4). Além disso, suportam a importação de diversos tipos de arquivos (requisito nº 5), ampliando as possibilidades de se utilizar outros *softwares* em conjunto, que forneçam insumos (objetos 3D, sons e imagens) para o desenvolvimento de ambientes.

Figura 8 – Visualização geral da estrutura de blocos definindo as interações do ambiente experimental.



Fonte: elaborado pelos autores.

Características fundamentais dos ambientes virtuais tridimensionais, animações e interações podem ser criadas sem que o designer tenha qualquer experiência anterior com programação computacional (requisito nº 6). O conjunto de programas traz consigo uma linguagem de blocos bastante intuitiva (**Erro! Fonte de referência não encontrada.**), com a qual é possível desenvolver facilmente animações e interações elaboradas de maneira predominantemente visual.

Outro importante ponto observado é que os dois *softwares* possuem um calendário regular de atualizações, corrigindo problemas encontrados e adicionando novas funcionalidades. Isso demonstra uma continuidade no suporte dado a eles e diminui as incertezas quanto a uma possível descontinuidade dos mesmos em um curto período de tempo (requisito nº 7). Por fim, ambos são inteiramente gratuitos, com direito a um grande e crescente número de recursos de qualidade comparada a *softwares* líderes de mercado, como 3ds Max por exemplo.

6 CONSIDERAÇÕES FINAIS

Inicialmente, a busca na bibliografia por informações acerca da representação tridimensional virtual e do WebGL possibilitou a definição de critérios e requisitos para a análise de 26 ferramentas encontradas. Verificando cada uma das ferramentas sob a ótica da aplicabilidade em projetos de Design Virtual, foi surpreendente encontrar um conjunto de *softwares* (Blender + Blend4Web) que satisfizesse inteiramente os requisitos estipulados, principalmente em um

universo de programas em que a grande maioria se mostrou ineficiente e com sérias limitações para trabalhar com as exigências atuais de qualidade para conteúdos 3D.

A experiência aprofundada de criação de um ambiente virtual tridimensional animado e interativo utilizando estes *softwares* exaltou mais ainda os benefícios trazidos pelo WebGL e que podem beneficiar os designers na produção e distribuição de seus projetos. Dentre as principais vantagens confirmadas por meio das análises e experimentações, estão:

- A visualização do conteúdo em multiplataforma (computadores, videogames, *smartphones* e *tablets*) sem a necessidade de complementos específicos para tal;
- As diferentes formas de interação suportadas;
- O ganho de qualidade gráfica em comparação a tecnologias predecessoras com propósitos semelhantes como o VRML e o X3D;
- A facilidade de exportação do conteúdo criado para um arquivo HTML executável;
- A familiaridade observada com as interfaces de *softwares* de modelagem 3D já utilizados no Design Virtual como o 3ds Max, por exemplo;
- O suporte a variados recursos de uso profissional para tratamento de geometrias, materiais, texturas, animações e interações;
- A disponibilidade de versões completas, até mesmo com licença de uso profissional, dos dois *softwares* de forma gratuita e para os principais sistemas operacionais do mercado (Windows, Mac OS e Linux).

Os pontos positivos ressaltados confirmam o cumprimento de todos os requisitos estipulados e a possibilidade de uma inserção eficiente da tecnologia WebGL, através dos *softwares* destacados, na atividade profissional de designers virtuais sem alterações significativas em seus métodos de trabalho, trazendo benefícios para a distribuição de seus projetos e atualização de suas capacidades e ferramentas. Também viabilizam a utilização da mesma para o ensino da criação de ambientes virtuais tridimensionais interativos. E por englobar todo o processo de desenvolvimento, da modelagem à exportação, se mostra como um bom vetor para o aprendizado de outros recursos úteis ao Design Virtual.

Para trabalhos futuros na área sugere-se a busca pela simplificação do processo de transmissão dos conhecimentos de linguagens de programação escrita aos designers, visando uma

ampliação ainda maior de suas competências e qualidade dos conteúdos 3D interativos criados. Recomenda-se também a atualização dos estudos aqui realizados e aplicações dos mesmos no desenvolvimento de outras aplicações como jogos virtuais, por exemplo.

REFERÊNCIAS

AGUILAR, Luis Joyanes . **Fundamentos de programação: algoritmos, estruturas de dados e objetos**. 3. ed. Porto Alegre: Mc Graw Hill, 2008. 706 p.

ANDRADE, Tiago. Bibliotecas de programação, o que podemos fazer com elas? In: INFINIT LOOP. Base de dados. 2015. Disponível em: <<http://www.infiniteloop.com.br/bibliotecas-de-programacao-o-que-podemos-fazer-com-elas/>>. Acesso em: 16 set. 2016.

ARORA, Sumeet. **WebGL Game Development**. Birmingham: Packt Publishing Ltd., 2014. 418 p.

COELHO, Tiago. **Linguagem de programação**. In: APLICAÇÕES INFORMÁTICAS B. Base de dados. 2011. Disponível em: <<http://eueovictor.blogspot.com.br/2011/05/linguagem-de-programacao.html>>. Acesso em: 16 ago. 2016.

CONGOTE, John et al. Interactive visualization of volumetric data with WebGL in real-time. In: INTERNATIONAL CONFERENCE ON 3D WEB TECHNOLOGY, 16., 2011, Paris. **Anais...** Paris: [s. n.], 2011. p. 137–146.

CUNHA, Thiago Monteiro Fernandes; MAINENTE, Cilene Aparecida. **Utilização de ambientes virtuais 3D no ensino de ciência da computação**: estado da arte. São Caetano do Sul: [s.n.], 2011. 15 p. Disponível em: <http://www.uscs.edu.br/pesquisasacademicas/images/pesquisas/thiago_cilene.pdf>. Acesso em: 08 out. 2018.

DANCHILLA, Brian. **Beginning WebGL for HTML5**. [S.l: s.n.], 2012. 356 p.

DINO. Divulgador de notícias. **Realidade Virtual**: "produtoras de conteúdo serão responsáveis pelo crescimento do mercado VR", diz diretor do VR Studios. In: EXAME. Base de dados. 2016. Disponível em: <<http://exame.abril.com.br/negocios/dino/noticias/realidade-virtual-produtoras-de-conteudo-serao-responsaveis-pelo-crescimento-do-mercado-vr-diz-diretor-do-vr-studios.shtml>>. Acesso em: 3 out. 2016.

FUPICAT. CN Animation. In: SCRATCH. Base de dados. 2016. Disponível em: <<https://scratch.mit.edu/projects/117341634/#editor>>. Acesso em: 08 out. 2018.

GUÉNO, François. Une sélection de startups de Futur en Seine. In: LE PHARE DIGITAL. Base de dados. 2014. Disponível em: <<http://lepharedigital.com/2014/06/23/selection-startups-futur-en-seine/>>. Acesso em: 08 out. 2018.

JÁCOME, Jarbas. **O que é biblioteca de programação | library | lib ? O que é API | Application Programming Interface ?** Blog no WordPress. [2010]. Disponível em: <<https://jarbasjacome.wordpress.com/o-que-e-biblioteca-de-programacao-library-lib-o-que-e-api-application-programming-interface/>>. Acesso em: 16 set. 2016.

LEE, Paul; STEWART, Duncan. **Virtual reality (VR): a billion dollar niche**. In: DELOITTE. Base de dados. 2016 Disponível em: <<https://www2.deloitte.com/tz/en/pages/technology-media-and-telecommunications/articles/tmt-pred16-media-virtual-reality-billion-dollar-niche.html>>. Acesso em: 18 out. 2016.

LEVKOWITZ, Haim; KELLEHER, Curran. Cloud and mobile web-based graphics and visualization. In: SIBGRAPI CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES TUTORIALS, 25., 2012, Ouro Preto: [s. n.], 2012. p. 21–35.

LITTLE Workshop. New Renault Espace. Base de dados. 2015. Disponível em: <<http://renaultespace.littleworkshop.fr/>>. Acesso em: 09 out. 2018.

LOESCH, Benedikt; CHRISTEN, Martin; NEBIKER, Stephan. Openwebglobe: an open source sdk for creating large-scale virtual globes on a webgl basis. In: ISPRS CONGRESS, 22., 2012, Melbourne. **Anais...** Melbourne: International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2012. p. 1-6. Disponível em: <<https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XXXIX-B4/195/2012/isprsarchives-XXXIX-B4-195-2012.pdf>>. Acesso em: 09 out. 2018.

MARINS, Vânia; HAGUENAUER, Cristina; CUNHA, Gerson. Realidade virtual em educação criando objetos de aprendizagem com VRML. **Revista Digital da CVA**, v. 4, n. 15, p. 1–11, 2007. Disponível em: <<http://pead.ucpel.tche.br/revistas/index.php/colabora/article/viewFile/97/83>>. Acesso em: 10 out. 2018.

MICROSOFT. **Entendendo quadros por segundo (FPS)**. Base de dados. [201-]. Disponível em: <<https://support.microsoft.com/pt-br/kb/269068>>. Acesso em: 10 out. 2018.

MINISTÉRIO DA CULTURA. **Conceito: Elevar o Nível de Abstração**. Base de dados. 2006. Disponível em: <http://mds.cultura.gov.br/core.base_rup/guidances/concepts/elevate_level_abstraction_7E5A12DB.html>. Acesso em: 16 set. 2016.

PAGET, Mat. Oculus Responds to Rift's Facebook Privacy Concerns. In: GAMESPOT. Base de dados. 2016. Disponível em: <<http://www.gamespot.com/articles/oculus-responds-to-rifts-facebook-privacy-concerns/1100-6438599/>>. Acesso em: 10 out. 2018.

PAN, Zhigeng et al. Virtual reality and mixed reality for virtual learning environments. **Computers and Graphics**, v. 30, n. 1, p. 20–28, 2006. Disponível em: <<https://ac.els->

cdn.com/S0097849305002025/1-s2.0-S0097849305002025-main.pdf?_tid=7105f2e3-1ecb-4081-a1a1-4ee5b76258c7&acdnat=1539188969_c75ce5ef52c219e72759f02b1fff9da3>. Acesso em: 10 out. 2018.

RUSSELL, Kenneth et al. Building 3D Web Applications using WebGL. In: ACM SIGGRAPH: Conference and Exhibition on Computer Graphics and Interactive Techniques, 8., 2015, Kobe. **Anais...** Kobe: [s.n.], 2015. 16 p.

SANTOS, Renato Manuel Simões. **Ensino da programação através de programação visual**. 2013. Trabalho apresentado na disciplina de Prática de Ensino, Universidade de Lisboa, Lisboa, 2013.

SIMPSON, Ronnie. **Firefox edges out Microsoft globally for first time in browser wars**. In: STAT COUNTERS. Base de dados. 2016. Disponível em: <<http://gs.statcounter.com/press/firefox-browser-edges-out-microsoft-globally-for-first-time>>. Acesso em: 7 fev. 2017.

VIRTUAL REALITY SOCIETY. **Augmented Reality: What is it?** Base de dados. [201-]a. Disponível em: <<http://www.vrs.org.uk/augmented-reality/>>. Acesso em: 23 nov. 2016.

_____. **What is Virtual Reality?** Base de dados. [201-]b Disponível em: <<http://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html>>. Acesso em: 21 jul. 2016.

ZHANG, Xiaoyu; GRAČANIN, Denis. An approach to WebGL based distributed virtual environments. In: INTERNATIONAL CONFERENCE ON 3D WEB TECHNOLOGY, 18., 2013, San Sebastian. **Proceeding...** San Sebastian: [s.n.], 2013. p. 195–198.