

Virtual Worlds and the 3D Web – time for convergence?

Hussein Bakri, Colin Allison, Alan Miller, Iain Oliver

School of Computer Science, University of St Andrews, United Kingdom
{hb, ca, alan.miller, iao}@st-andrews.ac.uk

Abstract. Multi-User Virtual Worlds (MUVW) such as Open Wonderland and OpenSim have proved to be fruitful platforms for innovative educational practice, supporting exploratory learning and generating true engagement. However, when compared with the way educational activities have flourished through the use of the constantly evolving WWW, MUVW learning environments remain a relatively obscure niche. Since the advent and promise of Second Life more than twelve years ago there has been no critical mass reached and no movement towards standardisation. Concomitantly, the 3D Web has emerged as a recognisable if loosely defined concept. With the advent of technologies such as WebGL and a plethora of plug-in 3D viewers for web browsers, the question arises: will MUVWs converge with the 3D Web? If so, can existing educational content be migrated to the 3D Web for mass dissemination? The paper contributes a survey of 3D Web and MUVW terms, concepts, technologies and projects, illustrating their similarities, their value for education and discusses the likelihood of convergence. The survey is complemented by a cultural heritage case study of Unity 3D support for the deployment of virtual worlds in web browsers using two different approaches.

Keywords: 3D Web, Web-Based Virtual World, Multi-User Virtual World, HTTP/2, VRML, X3D, X3DOM, WebGL, HTML-5, O3D, Oak3D, Unity 3D

1 Introduction

Multi-User Virtual Worlds (MUVW) such as Second Life [1], OpenSim [2] and Open Wonderland [3] are sophisticated client-server systems which have been used extensively for immersive learning activities [4][5][6]. These MUVW systems have their own schemes for communications, programming, graphics, rendering and control. By comparison the standard web consists of universally agreed standards and protocols such as HTTP, HTML and JavaScript, thereby creating a platform for interactive educational resources that can readily be made globally accessible. So although OpenSim notably describes itself as a “prototype 3D Web” - aiming to create a web of Virtual World Grids connected through portals allowing avatars to navigate seamlessly between virtual worlds that exist in different administrative domains – participation is confined to OpenSim-based sites which are connected to the OpenSim *hypergrid*. With the advent of the “3D Web” conceived of as a direct evolution of the standard web, what is the likelihood that the rich immersive experience of educational

MUVWs can be deployed across the standard web, delivered via standard browsers? This research seeks to identify and qualify signs of convergence between MUVWs and the 3D Web. In addition, a cultural heritage case study of the performance of a virtual world originally created in OpenSim but now capable of being run in standard browsers is included to provide performance insights into using a web browser as a virtual world viewer.

The paper is structured as follows: section 2 provides background on well-known 3D Web technologies. Section 3 is divided into three parts. The first part lists some Web-Based Viewers for existing well-known MUVWs and the second part discusses some projects aiming to create Web-Based Virtual Worlds. This section then discusses how both categories of environments can be used in education. Section 4 analyses how HTTP/2, the newest version of the web network protocol, is useful for supporting virtual world traffic for the 3D Web; Section 5 compares the performance of WebGL and Unity Web Player versions of a virtual world and Section 6 concludes.

2 3D Web Technologies

In this section we provide a brief review of the most commonly used 3D Web tools and languages.

2.1 VRML, X3D and X3DOM

Virtual Reality Modelling Language (VRML) [7] was the first technology for building and delivering 3D Web content. The language incorporates 2D and 3D graphics, multimedia and animation combined with powerful scripting and multiuser networking capabilities.

eXtensible 3D (X3D)[8] is a royalty-free standard which succeeded VRML. X3D advances include techniques such as programmable shaders, geo-location, multi-texturing and humanoid animations. Its XML encoding makes it easy to incorporate into web services leading to improved data compression and faster downloads over the web.

The X3DOM model [9] is X3D in the DOM tree. With this model, X3D can be fully integrated into any webpage in the same way as SVG (Scalable Vector Graphics) can. Nodes of X3D can be declared inside basic XHTML elements and scenes of X3D can be modified using JavaScript. The current implementation of X3DOM is based on WebGL. With the majority of browsers now supporting WebGL, X3DOM code can be embedded and manipulated without any plugin.

2.2 WebGL

WebGL (Web Graphics Library)[10] is a standard introduced by Khronos for web browsers. It consists of JavaScript Libraries accessing OpenGL ES (OpenGL Embedded System) a subset of OpenGL[11]. WebGL is rendered in the HTML-5 canvas element of a page. WebGL allows browsers to render 3D content without plug-ins.

Chrome Experiments [12] aim to prove and show the power of WebGL. They include sophisticated 3D navigation, physics engine and collision projects. Many

WebGL projects demonstrate sophisticated hardware accelerated 3D Web graphics rendered in real-time using WebGL, HTML5 and CSS3-3D. In addition, Mozilla Experiments using WebGL [13] show impressive game engine performance. This shows how the web is becoming suitable for 3D games and therefore, in principle, for web-based virtual worlds.

2.3 HTML-5 and the CSS3 3D features

In HTML-5 [14] the “canvas” element is the only facility for browsers to render bit-map images of 3D content without a plug-in. Solutions that don’t require plug-ins are mainly implemented through models and languages like X3DOM and WebGL [15]. There are Canvas3D HTML plug-ins for Firefox and Opera. The Firefox plug-in works the same way WebGL does by allowing the programmer to access the OpenGL ES library through JavaScript. 3D rendering can be hardware accelerated in the Firefox Canvas 3D HTML element. Opera developed their own 3D programming API as a wrapper over the powerful and stable graphics libraries of Open GL and DirectX [16]. HTML-5 browsers include features such as multi-threaded programming which can better support immersive virtual world graphics.

CSS3 3D transformations and animations are a way to enrich CSS rules with 3D capabilities. A web programmer has the ability to scale, rotate, skew, and change the perspective of almost any DOM element creating a complete 3D space using only CSS and HTML [17].

2.4 3D Markup Language for the Web (3DMLW)

3D Markup Language for the Web (3DMLW) is an open source XML-based language for presenting 2D and 3D dynamic and interactive content on the web. By using the Lua scripting language [18], programmers can animate complex 3D scenes. Plug-ins are available for different browsers together with several authoring tools [16].

2.5 Java 3D and Java OpenGL (JOGL)

Java3D is a runtime scene graph 3D API wrapper around OpenGL and DirectX, incorporating VRML/X3D. Sun stopped supporting the development of Java3D in mid-2003 and it became a community source project. In 2012, Java3D became Java OpenGL (JOGL) a low level interface providing binding to OpenGL libraries[16].

2.6 O3D and Oak3D

O3D [16][19] is a Google open source project originally requiring a plug-in. It works by using a JavaScript API for creating 3D dynamic and interactive content in a web browser. It exploits hardware accelerated graphics to support advanced rendering effects, detailed texturing, shadows and reflections, particle systems and physics engines in a web browser. O3D code can be edited in real time. Recent implementations [19] are built on top of WebGL and thus do not require plug-ins where a browser supports WebGL.

Oak3D [20] is an open source JavaScript library for creating interactive 3D content. It is implemented on top of HTML-5 and WebGL and like the recent version of O3D does not require a plug-in.

2.7 Web-generated builds from Game Engines

Unity 3D and Unreal are two of the most popular general purpose game engines. They can generate web builds for games and 3D worlds which can then be accessed through a plug-in or through HTML-5 and WebGL. Other games engines such as ImpactJS [21] and CreateJS [22] are tailored specifically for games on the web.

Unity 3D. Unity3D [23] supports the development of 3D games and complex 3D environments on a variety of platforms and devices including web browsers either through a plug-in or as WebGL. The engine has a highly optimized graphics pipeline for OpenGL and DirectX and supports many features including advanced meshes, advanced lighting and shadows, particle systems and the Ageia PhysX physics engine. Section 5 reports on the relative performance characteristics of a virtual world originally hosted in OpenSim, then transformed into Unity, then generated in two formats suitable for web browsers.

Unreal Engine. Unreal Engine [24] is developed by Epic Games. It is written in C++ which is also used as the main game programming language. The Blueprint visual scripting language with debugging capabilities can also be used. Unreal engine also supports generating a WebGL format of a game for HTML-5 browsers.

2.8 The 3D Web and e-learning

In medical education, Birr et al.[25] developed a real-time Web surgical teaching tool derived from patient specific image data. The tool was developed using X3D and WebGL. Landro et al. [26] developed a web portal containing 3D objects simulating highly interactive virtual laboratories. In Entomology (the study of insects), X3D models are created to represent high quality natural-colour detailed 3D models of various insects [27]. 3D Web virtual labs in engineering are quite numerous. Violente et al.[28] designed web-based interactive 3D concept maps for an engineering drawing course and showed that this helped students with deficits in spatial abilities to better represent the content to be learned. SAFAS (Structure And Form Analysis System) [29], is a 3D Web design tool for architects and structural engineers. It simulates real forces, loads, displacements and deformations of 3D complex structures of steel nodes and members. In Mathematics, Hennig et al. [30] developed a blended learning scenario based on abundant use of 3D visualizations of mathematical concepts using WebGL. Raman et al.[31] Implemented a 3D convex glass lens experiment in Olab Physics using WebGL and Dynamic cube mapping. In cultural heritage education, Wang et al. [32] developed a game-based virtual environment for learning cultural heritage. The environment is built using the O3D API.

3 Web-Based Viewers & Web-Based Virtual Worlds

In this section, we present a brief overview of some of Web-Based Virtual Worlds (WBVWs) that are complete 3D spaces on the web and survey Web-Based Viewers (WBV) for existing MUVWs. WBVs offer several benefits over standalone MUVW clients in educational settings. These include:

- WBV function better in educational organizations because they use standard web protocols such as HTTP thus avoiding the problem of MUVW firewall blocking discussed in Section 5.
- No need for stand-alone client installation thus making these environments easily accessed in education due to the fact that many institutions have strictly controlled computer laboratories.
- Unanimity of quality through web browsers and the suitability of these environments for regular web users with no technical expertise.
- WBVs present a means of access from mobile devices with limited resources (only a browser is needed) thus promoting pervasiveness and ubiquity.

3.1 Web-Based Viewers for MUVWs

Web-Based Viewers are alternatives to the traditional stand-alone MUVW viewers. A small selection is described below.

Unity3D based Virtual Viewer. Katz et al. [33] developed a Unity3D plug-in Virtual Viewer that connects to MUVWs and renders Second Life and OpenSim scenes.

Project Skylight - Second Life Web-Based viewer. Project Skylight[34] was a beta testing web viewer for Second Life that was launched in November 2010. A trial offered selected visitors 60 minutes to explore Second Life inside their web browsers (no need to download the viewer at all). Second Life has also used the Cloud-based rendering and streaming game service Gaikai [35] in *Skylight*. Linden Labs wanted to use this tool to evaluate whether Second Life should invest in a Web-Based Viewer.

TipoDean Web Viewer. TipoDean[36] offers a conversion tool that transforms content developed in OpenSim and Second Life into Unity3D. Spatial VOIP is also supported.

3DXplorer viewer. 3DXplorer [37] is an interactive 3D viewer created and maintained by Altadyn that allows users to design 3D immersive virtual worlds in web browsers both on desktop and mobile devices requiring no plug-in but requires the presence of a Java Virtual Machine.

Cube3. Cube3 [38] is a Unity3D based tool that allows access and usage of a set of modular/customized virtual worlds builds and environments specifically designed for educational and collaborative use. These virtual spaces are integrated with Second Life and OpenSim prim based systems. The environments contain buildings and fur-

niture customizable under the demand of the user. These environments are accessed from any web browser and from mobile devices like iPhones.

Pixie Viewer. Pixie Viewer[39] is a browser based viewer for 3D virtual worlds. It runs on any modern HTML-5/WebGL browser including Safari, Chrome, Firefox, Opera and others. The viewer can be either connected to a standalone backbone or to OpenSim grids although this requires a special module to be available in OpenSim. Pixie Viewer has a set of building and texturing tools for users. It also has a 3D printing facility from the viewer itself for objects.

3.2 Web-Based Virtual Worlds

This section provides some examples of projects building virtual worlds on the web. Web-Based Virtual Worlds appear to be completely integrated into the web from the perspective of the user. All the user has to do is to navigate the environment from any web browser.

KataSpace. Kataspace [40][41] is an open source project which runs on top of WebGL and HTML5. KataSpace is built using the Sirikata [42] multiuser platform for games and virtual worlds.

Virtual World Framework. Virtual World Framework (VWF) [43] is a means to connect virtual worlds and 3D entities and content via web browsers. It is an open source platform that allows anyone to build collaborative 3D applications on the web. It also extends and interfaces with existing client-server virtual worlds like OpenSim making it possible for these environments to be delivered in a lightweight manner via web browsers. VWF uses WebSockets and WebGL.

Sandbox [44] is a Virtual World Framework (VWF) authoring and delivery platform for creating 3D Environments on the web. Users can create complex and beautiful virtual worlds on the web at no cost. The tool uses WebGL and does not need a plugin.

ReactionGrid Jibe platform based on Unity3D. ReactionGrid Inc. [45] is company specialising in the creation of 3D Multi-User Virtual Environments on the web and on mobile devices. ReactionGrid Inc. uses Jibe, a platform based on a Unity3D plugin in web browsers. Jibe enables users to create and manage their own complex virtual environments incorporated and accessible through the web. Jibe is also integrated into Facebook through an application and can run on Android and iOS systems.

Cloud Party - a virtual 3D space in browsers. Cloud Party [46] is a 3D virtual world that runs in Firefox and Chrome. The company behind it, Cloud Party Inc, has been acquired by Yahoo. The environment is integrated into Facebook and enables creating, editing and uploading of 3D objects. Users can customize avatars, land and sky, and can script objects for certain effects [47].

Altadyn Web-Based Virtual World projects. Altadyn's products [48] allow users to access 3D spaces on the web for collaborative purposes thus bridging the gap between 3D virtual worlds and web conferencing. It works on browsers including Safa-

ri, Firefox and Internet Explorer without a plug-in but a Java Virtual Machine is needed. The 3D virtual products and services are based on JOGL and Lightweight Java Game Library (LWJGL) technologies for providing real time rendering. Environments like Business Hangouts, 3d-virtualevents, 3d-virtualembassy are examples of complex collaborative virtual worlds on the web. Altadyn WBVWs are convenient for virtual lectures to a large audience of students.

Avaya web.alive. Avaya web.alive [49][50] is based on the Unreal Engine and 3D Web technologies. It provides a virtual environment used for collaboration, product presentations and business meetings which can be accessed from any web browser. It allows users to present documents in common formats (PPT, PDF), to stream videos, to share files, to share desktops and to use live webcams with HD audio. Avatars can be customized to suit the needs of users. Avaya has a 3D content development toolkit for editing the environment and creating content. It allows for importing models created in Autodesk product and Blender.

RealXTend Tundra [51][52] is an open source project that allows users to create 3D virtual worlds and video games. It has an open source client called Naali. The client can also connect to Second Life and OpenSim platforms. RealXTend is implemented using the Tundra SDK and has an add-on for OpenSim. The viewer Naali and RealXTend Tundra server use HTTP and XMPP (Extensible Messaging and Presence Protocol).

Meshmoon [53], created by Adminotech Ltd., is a virtual world platform based on realXtend Tundra. It also has a 3D application hosting service which uses cloud storage and is designed to allow users to share 3D virtual worlds online. The worlds can be populated by 3D meshes imported from 3D software packages like Blender, 3DS Max, Maya among others. It is also possible to import content from PDF, web pages, pictures, media players and presentation tools into frames in the 3D Scene. Developers can use scripting to give different functionalities to 3D models. The service promises scalability and accessibility to users. Meshmoon provides communications through both VOIP chat via WebRTC[54] and text chat. Meshmoon is accessible through a web-based viewer called Web Rocket which uses WebGL and WebSockets and through a standalone application called the Rocket Client.

Open Virtual World Web Projects. Unity3D version 5 delivers WebGL content directly to a browser, obviating the need for a plug-in. Preliminary research by the Open Virtual Worlds group [55] has shown that it is possible to capture OpenSim texture packets and convert them to Unity 3D which can then be delivered to Firefox or Chrome via WebGL. Educational projects originally hosted in OpenSim/Second Life such as the Laconia Acropolis Virtual Archaeology (LAVA) [56] and Timespan Longhouse [57] have been transformed into Unity 3D. A performance study of two web versions of the Timespan Longhouse in Chrome is presented in Section 5.

4 HTTP/2 and QUIC for Virtual Worlds and the 3D Web

HTTP/2 [58] and QUIC [59] protocols could be used for MUVWs and the 3D Web [60]. In most educational settings firewall port blocking inhibits the use of Second Life / OpenSim as they require a range of “unusual” UDP ports to be opened for effective client-server communication [61]. At the same time, MUVW traffic has soft real time requirements and is sensitive to delay, jitter and bandwidth constraints – hence the original Second Life communication model which has a complex set of UDP-based *circuits* and *channels*. HTTP/2 has the potential to offer the same functionality through two TCP ports, 80 & 443, which are normally open. QUIC [59] can be used in place of HTTP/2 but runs over UDP instead of TCP. QUIC uses port 80 for HTTP and port 443 for HTTPS and uses the same “HTTP” and “HTTPS” URI schemes. See [60] for further information how they could be used in virtual worlds. Features of value that are not present in HTTP 1.1 include:

- **Request and Response multiplexing:** OpenSim traffic is divided into channels for Assets, Textures, Task, Wind [62]. Multiplexing of streams in HTTP/2 or QUIC could be used to multiplex these channels.
- **Streams can be interleaved and prioritized:** Studies [62][63] show that certain type of virtual world traffic needs to have some precedence e.g. avatar control information. This leads to a better user Quality of Experience (QoE). Prioritization of HTTP/2 or QUIC streams could be used to solve this problem.
- **Server Push:** This allows a virtual world server to anticipate what a client needs and send data that might be needed later without waiting for the client to ask for it e.g. neighbouring region information.

5 Case Study: Unity 3D Builds for Web browsers



unity WebGL

Timespan-Longhouse1

Figure 1: The WebGL build of the Timespan Longhouse

The Timespan Longhouse (TL) [57], is part of a larger virtual world hosted originally in OpenSim and then transformed into Unity 3D by the Open Virtual Worlds group [55]. Unity can generate two types of build suitable for browsers: Unity Web Player (UWP) and WebGL (U-WebGL). The crucial difference is that UWP requires a plug-in to be installed whereas U-WebGL can be run directly in any browser that supports WebGL, such as Firefox or Chrome. A U-WebGL view of the TL displayed in Chrome is shown in Figure 1. We focus on two key QoE metrics for comparison: frames per second (FPS) and initial download time (IDT) - a 3D Web equivalent of the widely used Page Load Time (PLT) metric in the 2D Web. FPS and Frame Time are important for the user experience in real time immersive environments. In an OpenSim viewer each frame should complete in approximately 18.18ms (55 frames per second). “If total frame time is greater than this then simulator performance will be degraded”[64]. Similarly, in Unity3D web applications, 60 FPS is recommended.

Notched Boxplot of Distributions of FPS

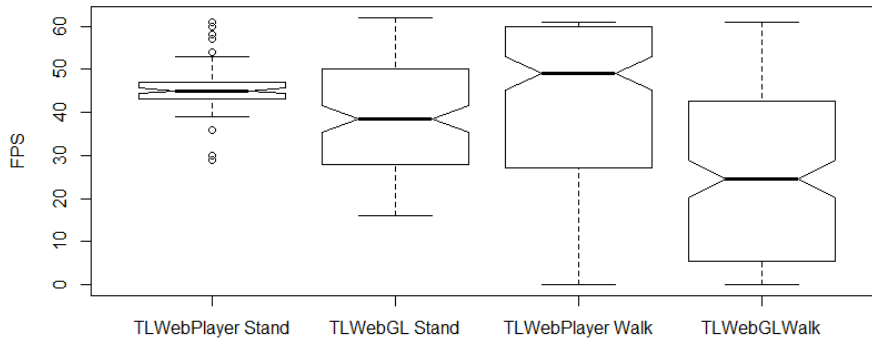


Figure 2: Frames per Second (FPS) in the two versions

Figure 2 summarises results from experiments where two mobility models were used:

- Standing: Avatar remains standing still with continuous yawing for 2 minutes (Yaw is the change in avatar orientation or the change in the “look at” view).
- Random Walking: Avatar randomly walks for 3 minutes in different directions (from non-dense to dense areas) and with a constant speed.

Figure 2 shows that the UWP build performed better (i.e. higher FPS) than the U-WebGL build, especially when the avatar is walking.

Timespan Longhouse	Average	Standard Deviation
UWP	3934.33 ms	91.26
U-WebGL	20191.67 ms	581.46

Table 1: Initial Download Times of Timespan Longhouse to Chrome

Table 1 shows that a user visiting the TL for the first time while using Chrome or Firefox would have to wait 20 seconds for the WebGL virtual world to download compared with 4 seconds for the UWP version. Neither is an impressive figure com-

pared with 2D Web expectations, where a Page Load Time of more than 1 second is considered poor. Further details about this study can be found in [65].

The experiments were performed on Chrome 44 which supported Unity Web player through NPAPI (Netscape Plugin Application Programming Interface). However, in early 2016 Chrome dropped support for NPAPI based plug-ins in version 45 and onwards [66]. Other plug-ins were also affected by this strategic decision to start moving towards a plugin-less browser. The Unity plug-in version is still supported at time of writing by browsers that still support NPAPI including Firefox, Opera and Internet Explorer but this is likely to change in the future e.g. Mozilla intends to remove support for NPAPI plug-ins. In the latest Unity beta version (5.4) the UWP build option has been removed implying that Unity is moving towards a WebGL only solution. The results of our experiments showed that UWP versions perform significantly better than WebGL versions, and it will be interesting to see if Unity can produce more efficient WebGL to address this problem, whose causes are partly explained in the following section.

Unity WebGL Limitations: WebGL is supported by the majority of web browsers to some degrees (e.g. cursor locking and full screen are not supported in Safari yet). There are some limitations in WebGL itself. JavaScript does not support multithreading or direct access to IP sockets of browsers for security concerns. WebGL supports only baked Global Illumination and not real-time ones. In addition, it does not support procedural materials or linear color rendering. It supports basic Web audio API which has many limiting audio features compared to other platforms (ex: no microphone class). The initial download time of a Unity WebGL world is a lot bigger than that of its plugin counterpart of the same world due to build sizes being a lot bigger. Another concern is the memory used by Unity WebGL builds. First the heap memory which Unity uses to store loaded assets and scenes needs to be at a convenient size to fit all data required to play the content in browser. Tweaking the WebGL heap memory size avoids many out-of-memory problems encountered normally in big WebGL builds. Another issue related to memory in WebGL is the memory used by the JavaScript engines for parsing and optimizing WebGL codes in browsers. Compression and optimization techniques of Unity WebGL builds minimize the emitted JavaScript code and thus result in smaller download times and lesser memory consumption [67]. WebGL 2.0 (based on OpenGL ES 3.0) mitigates some of the limitations of WebGL 1.0 but is still experimental and is not yet supported in the majority of browsers [67].

6 Conclusion

Figure 3 shows key markers in Virtual World and 3D Web developments that have been discussed in this paper. We have shown that many 3D Web technologies and tools - especially WebGL – can now host virtual worlds similar to those built in OpenSim and Open Wonderland, and that traditional web browsers can even act as viewers for some of the existing MUVWs. Following this trend of developments it is possible that there will be a point in the future where we will see fully fledged virtual worlds on the web that have all the features currently present in MUVWs.

From the educational perspective we see that web-based learning environments are increasingly accommodating multi-user interactive 3D formats with the help of 3D Web technologies. This convergence also allows for some of the existing MUVW immersive educational content to migrate to the 3D Web where it is much more accessible.

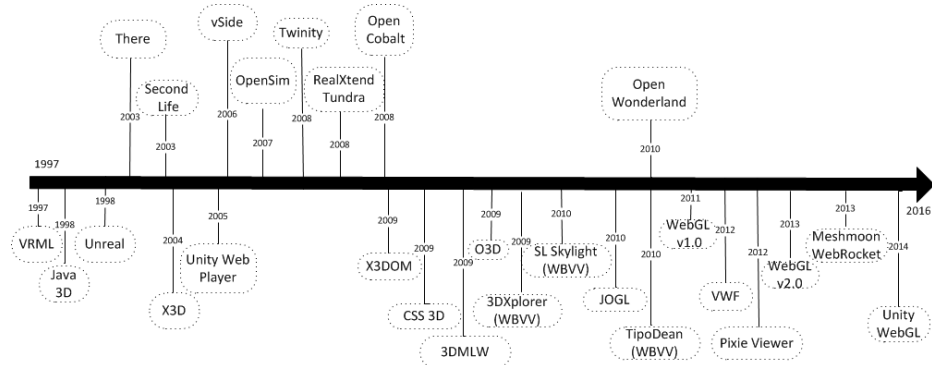


Figure 3: Markers in the development of virtual worlds and the 3D Web

We have posited that the most recent web network protocols, HTTP/2 and QUIC, would be useful in supporting real-time 3D traffic for both MUVWs and WBVWs, and avoid the problems encountered with restrictive campus firewalls and the use of non-standard transport protocols.

A small case study, taken from a larger set of experiments, has shown that the UWP version of the Timespan Longhouse, an educational cultural heritage project, performed significantly better than the WebGL version. However, with the move evidenced in Chrome and Mozilla towards plugin-less browsers Unity is dropping UWP in favour of WebGL which may result in better optimized output.

In summary there are many signs of convergence but it is still not possible to state categorically that direct support for immersive environments in the standard web will increase user demand which in turn will spur further and faster innovation in web-based immersive education.

References

- [1] "Second Life Main Website." [Online]. Available: <http://secondlife.com/>. [Accessed: 28-Mar-2015].
- [2] "OpenSim Official Website." [Online]. Available: http://opensimulator.org/wiki/Main_Page. [Accessed: 02-Apr-2015].
- [3] "Open Wonderland Official Website." [Online]. Available: <http://openwonderland.org/>. [Accessed: 01-Apr-2015].
- [4] I. Perera, C. Allison, J. R. Nicoll, and T. Sturgeon, "Towards successful 3D virtual learning - a case study on teaching human computer interaction," *2009 Int. Conf. Internet Technol. Secur. Trans.*, 2009.
- [5] T. Sturgeon, C. Allison, and A. Miller, "802.11 Wireless Experiments in a Virtual

- World,” *ACM SIGCSE Bull.*, vol. 41, no. 3, p. 85, Aug. 2009.
- [6] J. Mccaffery, A. Miller, and C. Allison, “Extending the use of virtual worlds as an educational platform - Network Island : An Advanced Learning Environment for Teaching Internet Routing Algorithms,” in *CSEDU 2011 : Proceedings of the 3rd International Conference on Computer Supported Education*, 2011, pp. 279–284.
- [7] Rickk Carey and Gavin Bell, “The annotated VRML 2.0 reference manual,” *Addison-Wesley Longman Ltd.* Addison-Wesley Longman Ltd., Essex, UK, 1997.
- [8] “What is X3D.” [Online]. Available: <http://www.web3d.org/x3d/what-x3d>. [Accessed: 16-Feb-2016].
- [9] “X3DOM.” [Online]. Available: <http://www.x3dom.org/>. [Accessed: 22-Feb-2016].
- [10] “WebGL - OpenGL ES 2.0 for the Web.” [Online]. Available: <https://www.khronos.org/webgl/>. [Accessed: 16-Feb-2016].
- [11] “OpenGL.” [Online]. Available: <https://www.opengl.org/>.
- [12] “Google Chrome Experiments using WebGL.” [Online]. Available: <https://www.chromeexperiments.com/webgl>. [Accessed: 22-Feb-2016].
- [13] “Mozilla - Demos of open web technologies.” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Demos_of_open_web_technologies. [Accessed: 21-Feb-2016].
- [14] M. Pilgrim, *HTML5: Up and Running*. “O’Reilly Media, Inc.,” 2010.
- [15] J. Yang and Z. Jie, “Towards HTML 5 and interactive 3D graphics,” *ICEIT 2010 - 2010 Int. Conf. Educ. Inf. Technol. Proc.*, vol. 1, no. Iceit, pp. 522–527, 2010.
- [16] P. Paulis, “3D Webpages,” *Študentská Ved. Konf. FMFI UK, Bratislava*, pp. 316–327, 2010.
- [17] B. Frain, *Responsive Web Design with HTML5 and CSS3*. Packt Publishing Ltd, 2012.
- [18] “Lua programming language main website.” [Online]. Available: <http://www.lua.org/>.
- [19] “O3D main page on Google Code.” [Online]. Available: <https://code.google.com/p/o3d/>. [Accessed: 22-Feb-2016].
- [20] “Oak3D.” [Online]. Available: <http://www.effecthub.com/t/oak3d>. [Accessed: 16-Feb-2016].
- [21] “Impact JavaScript Game engine.” [Online]. Available: <http://impactjs.com/>. [Accessed: 22-Feb-2016].
- [22] “CreateJS.” [Online]. Available: <http://createjs.com/>. [Accessed: 22-Feb-2016].
- [23] “Unity3D game engine.” [Online]. Available: <http://unity3d.com/>.
- [24] “Unreal Engine 4.” [Online]. Available: <https://www.unrealengine.com/what-is-unreal-engine-4>. [Accessed: 22-Feb-2016].
- [25] S. Birr, M. Jeanette, D. Sommerfeld, and B. Preim, “A novel Real-Time Web3D Surgical Teaching Tool based on WebGL,” *Bild. für die Medizin 2011*, 2011.
- [26] T. di Mascio, R. Gennari, A. Melonio, and P. Vittorini, “A Collaborative Distance Learning Portal Integrating 3D Virtual Labs in Biomedicine,” *Adv. Intell. Syst. Comput.*, vol. 292, pp. 135–142, 2014.
- [27] M. Adcock, C. Nguyen, D. Lovell, and J. La Salle, “Accelerating Entomology with Web3D Insects,” p. 4503, 2014.
- [28] M. G. Violante and E. Vezzetti, “Design of web-based interactive 3D concept maps: A preliminary study for an engineering drawing course,” *Comput. Appl. Eng. Educ.*, vol. 23, no. 3, pp. 403–411, 2015.

- [29] N. Polys, F. Bacim, M. Setareh, and B. Jones, "Building novel Web3D user interfaces: a case study from architecture and structural engineering," *Proc. 18th Int. Conf. 3D Web Technol.*, pp. 135–138, 2013.
- [30] M. Hennig, D. Gaspers, and B. Mertsching, "Interactive WebGL-based 3D visualizations for situated mathematics teaching," *2013 12th Int. Conf. Inf. Technol. Based High. Educ. Training, ITHET 2013*, pp. 1–6, 2013.
- [31] R. Raman, U. Kv, S. Rekha V, and P. Nedungadi, "Using WebGL to implement a glass lens in Online Labs," *2014 7th Int. Conf. Contemp. Comput. IC3 2014*, pp. 50–55, 2014.
- [32] L. Wang, J. W. Guo, C. L. Yang, H. S. Zhao, and X. X. Meng, "O3D-based game learning environments for cultural heritage online education," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6249 LNCS, pp. 417–428, 2010.
- [33] N. Katz, T. Cook, and R. Smart, "Extending Web Browsers with a Unity 3D-Based Virtual Worlds Viewer," *IEEE Internet Comput.*, vol. 15, no. 5, pp. 15–21, 2011.
- [34] "Beta Testing of Project Skylight Web Viewer for Second Life." [Online]. Available: <http://ambrosiadanceclub.com/news/beta-testing-of-skylight-webviewer->. [Accessed: 21-Feb-2016].
- [35] "Gaikai." [Online]. Available: <https://www.gaikai.com/>. [Accessed: 21-Feb-2016].
- [36] "Tipodean Main Website." [Online]. Available: <http://www.tipodean.com/>. [Accessed: 21-Feb-2016].
- [37] "3DXplorer." [Online]. Available: <https://3dexplorer.wordpress.com/>. [Accessed: 21-Feb-2016].
- [38] "Cube3 Main Website." [Online]. Available: <http://www.cube3.com/>. [Accessed: 21-Feb-2016].
- [39] "Pixie Viewer for OpenSim." [Online]. Available: <http://pixieweviewer.com/>. [Accessed: 01-Apr-2015].
- [40] "KataSpace Application Built on Sirikata." [Online]. Available: <http://www.sirikata.com/blog/2010/kataspace-application-built-on-sirikata/>. [Accessed: 22-Feb-2016].
- [41] "Github Kataspace Repository." [Online]. Available: <https://github.com/sirikata/kataspace>. [Accessed: 22-Feb-2016].
- [42] "Sirikata." [Online]. Available: <http://www.sirikata.com/>. [Accessed: 21-Feb-2016].
- [43] "Virtual World Framework Main Website." [Online]. Available: <https://virtual.wf/documentation.html>. [Accessed: 01-Apr-2015].
- [44] "The Virtual World Framework Sandbox." [Online]. Available: <https://sandbox.adlnet.gov/110/adl/sandbox/>. [Accessed: 21-Feb-2016].
- [45] "ReactionGrid Main Website." [Online]. Available: <http://reactiongrid.com/>. [Accessed: 03-Apr-2015].
- [46] "Cloud Party Main Website." [Online]. Available: <http://www.cloudparty.com/>.
- [47] İ. G. Aydın, "Comparison of two Virtual Worlds based on their pedagogical affordances and constraints as interactive learning environments," no. 2008, 2013.
- [48] "ALTADYN Corp." [Online]. Available: <https://business-hangouts.com/>. [Accessed: 20-Feb-2016].
- [49] "AvayaLive Engage Web Site." [Online]. Available:

- <https://engage.avayalive.com/Engage/>. [Accessed: 22-Feb-2016].
- [50] “Benchmark of 3D Virtual Environments.” [Online]. Available: https://blogs.aalto.fi/allthingsvirtual/files/2011/06/Benchmark_Report1.pdf.
- [51] T. Alatalo, “An Entity-Component Model for Extensible Virtual Worlds,” *IEEE Internet Comput.*, vol. 15, no. 5, pp. 30–37, 2011.
- [52] “Tundra.” [Online]. Available: <https://github.com/realXtend/tundra/wiki>. [Accessed: 26-Feb-2016].
- [53] “Meshmoon Documentation.” [Online]. Available: <http://doc.meshmoon.com/index.html>. [Accessed: 26-Feb-2016].
- [54] “WebRTC.” [Online]. Available: <https://webrtc.org/>.
- [55] “Open Virtual Worlds.” [Online]. Available: <http://www.openvirtualworlds.org/>.
- [56] K. Getchell, A. Miller, J. R. Nicoll, R. J. Sweetman, and C. Allison, “Games methodologies and immersive environments for virtual fieldwork,” *IEEE Trans. Learn. Technol.*, vol. 3, no. 4, pp. 281–293, 2010.
- [57] J. McCaffery, A. Miller, S. Kennedy, T. Dawson, C. Allison, A. Vermehren, C. Lefley, and K. Strickland, “Exploring Heritage through time and space supporting community reflection on the highland clearances,” *2013 Digit. Herit. Int. Congr.*, vol. 1, pp. 371–378, 2013.
- [58] M. Belshe, R. Peon, and M. Thomson, “RFC 7540: hypertext transfer protocol version 2 (HTTP/2).” Tech. Rep., May 2015. [Online]. Available: <https://tools.ietf.org/html/rfc7540>, 2015.
- [59] “QUIC Protocol Official Website.” [Online]. Available: <https://www.chromium.org/quic>. [Accessed: 03-Apr-2015].
- [60] H. Bakri, C. Allison, A. Miller, and I. Oliver, “HTTP/2 and QUIC for Virtual Worlds and the 3D Web?,” *Procedia Comput. Sci.*, vol. 56, FNC, pp. 242–251, 2015.
- [61] C. Allison, A. Campbell, C. J. Davies, L. Dow, S. Kennedy, J. P. McCaffery, A. H. D. Miller, I. A. Oliver, and G. I. U. S. Perera, “Growing the use of Virtual Worlds in education : an OpenSim perspective,” *EiED 2012 Proc. 2nd Eur. Immersive Educ. Summit*, pp. 1–13, 2012.
- [62] I. A. Oliver, A. H. D. Miller, and C. Allison, “Virtual worlds, real traffic,” *Proc. first Annu. ACM SIGMM Conf. Multimed. Syst. - MMSys '10*, p. 305, 2010.
- [63] I. Oliver, A. Miller, and C. Allison, “Mongoose: Throughput redistributing virtual world,” *2012 21st Int. Conf. Comput. Commun. Networks, ICCCN 2012 - Proc.*, 2012.
- [64] “OpenSim Client Side Monitoring.” [Online]. Available: http://opensimulator.org/wiki/Client_side_monitoring. [Accessed: 05-Feb-2016].
- [65] H. Bakri and C. Allison, “Measuring QoS in Web-Based Virtual Worlds: an evaluation of Unity 3D web builds,” in *MMSys '16 Multimedia Systems Conference 2016 Klagenfurt am Wörthersee, Austria — May 10 - 13, 2016*, ACM Press - Association for Computing Machinery, 2016.
- [66] “Unity Web Player Roadmap.” [Online]. Available: <http://blogs.unity3d.com/2015/10/08/unity-web-player-roadmap/>. [Accessed: 03-May-2016].
- [67] “Unity Documentation.” [Online]. Available: <http://docs.unity3d.com/Manual/index.html>. [Accessed: 01-Sep-2015].