WebGL on Mobile Devices

William Almnes, *University of Oslo*
Marko Andjic, *University of Oslo*
Matthias Armbruster *University of Mannheim*
and Paul Steinhilber, *University of Mannheim*

Abstract—The abstract goes here.

Index Terms—WebGL, mobile computing, teleseminar.

1 Introduction

The technologies which allow people to interact with the world around them evolve constantly. Whereas it was common to simply display information in a static form in the early days of the web, at present an ever increasing amount of content is presented in dynamic and interactive ways.

In the effort to expand internet browser functionality to natively support three-dimensional graphics, HTML5, WebGL have been developed. Many web sites already offer functionalities which were previously only found in native applications, e.g. word processing using Google Docs [1] or creating presentation using 280 Slides [2], thus narrowing "the gap between them" [3]. A native usage of 3d-functionality enhances these "web applications" even further.

"Augmented Reality (AR) has been defined broadly as combining real and computer-generated digital information into the users view of the physical and interactive real world in such a way that they appear as one environment, thus providing a bridge between digital information and the physical world" ([4], [5], [6], [7] via [8]). Application areas of AR can be as broad as ranging from health care [9] to education [10] to tourism [11]. The topic has gained momentum thanks to the rise in smartphone usage. "To make the world itself the user interface [...] may revolutionize the way information is accessed and presented to people in the future [4], [7].

The rise of smartphones is growing with a fast and still accelerating pace, enabled ways of displaying information in a new way in a truly mobile context to many people [12], [13]. They offer a much higher power than feature phones and even claim to offer the "real web" experience with "real browsers" [14]. There are differences, however, between the way information can be accessed from a desktop system and a mobile device, influenced by factors like screen size, processing power, and input methods.

3d-support on mobile devices is still in an early phase. Adobe abandoned Flash in late 2011 [15], leaving WebGL as the main (method) of providing interactive 3d content on the mobile web, even if it is still in an initial phase.

This paper analyzes the status quo and potentials of WebGL on mobile devices (regarding performance and human-computer-interaction), and answers the following questions:

1) **Are Augmented Reality applications possible on mobile devices using only**

- *William W. F. Almnes: wwalmnes@student.matnat.uio.no*
- *Marko Andjic: marko.andjic@usit.uio.no*
- *Matthias Armbruster: marmbrus@rumms.uni-mannheim.de*
- *Paul R. E. Steinhilber: ps@paulsteinhilber.de*
- *Daniel Schön: schoen@informatik.uni-mannheim (advisor)*

Submitted just before April 23, 2012.

the browser?

2) **Is 3d-augmentation of these applications possible using WebGL?**
3) **How fast is WebGL on mobile devices?**

The paper is structured as follows: Chapter 2 gives background information on relevant topics, i.e. augmented reality, WebGL, and human-computer-interation (HCI) with mobile devices and their evaluation criteria; Chapter 3 gives implementation details of the the WebGL environment used to capture the status quo; Chapter 4 presents the evaluation as well as limitations to this study; and Chapter 5 gives a summary of the results and presents future research opportunities.

2 Background

2.1 WebGL

In 2009, the *WebGL Working Group* was formed with the mission to ”bring hardware-accelerated graphics to the internet” [16]. WebGL is a software library and extends Javascript to allow it to natively generate websites with 3D content. The first version of WebGL was released in 2011 [17].

2.1.1 Design

”WebGL is based on OpenGL ES 2.0 and provides an API for 3D graphics. It uses the HTML5 canvas element and is accessed using Document Object Model interfaces. Automatic memory management is provided as part of the JavaScript language.... Renders of GPU, which muss support shader rendering” (Wikipedia) It is based on Canvas 3D, which was developed at Mozilla [18]. ”WebGL is a context of the canvas HTML element that provides a 3D computer graphics API without the use of plug-ins.” (Wikipedia)

2.1.2 Desktop WebGL

2.1.3 Mobile WebGL

Although support for WebGL is available on the desktop computers within all major browsers WebGL (FireFox 4+, Safari 5.1+, Chrome 10+, Opera 12+), except of Microsoft’s Internet Explorer [3], browsers on mobile devices supporting WebGL are still very rare.

Apple has added WebGL capabilities to iOS with iOS 4.2 [19], [20]. However officially WebGL is only available to be used on Apples iAd platform [19], [21]. With an hack discovered by Nathan de Vries [19] WebGL can also be enabled in a UIWebView. Using this way it is possible to build a custom WebGL viewer for iOS. However since this method of enabling WebGL on iOS requires the use of non-public APIs it cant be made available to end users due to Apple’s App Store Review Guidelines [22]. The built-in Safari browser on iOS does not support WebGL and due to Apples App Store Review Guidelines [22] third party browsers are required to use the WebKit rendering engine provided by Apple. So until Apple officially supports WebGL in Safari on iOS there is no way available for the end user to run WebGL content on iOS.

As the desktop version of InternetExplorer does not support WebGL [3] and Microsoft is considering WebGL ”harmful” [23] we probably wont see WebGL support on Windows Phone 7 in the near future.

With Firefox 4 a WebGL compatible browser is available on Android.

Also RIM’s tablet, the BlackBerry PlayBook, supports WebGL in web applications.

2.1.4 Security Issues

Although the support of WebGL seems to be widespread, there are also critical voices, with Microsoft probably being their most prominent spokesperson. Microsoft believes ”that WebGL will likely become an ongoing source of hard-to-fix vulnerabilities” [23], because WebGL allows direct access to the computers hardware from the web. WebGL security therefore relies on graphic card drivers and other third party components. [23]

Context, an information security consultancy, which recommends to disable WebGL in the browser, published and demonstrated two possible attack scenarios both initiated from a malicious website [24], [25]. They were not only able to perform a successful Denial of Service attack, which leads to operating system crashes and freezes of the desktop, but also to gather possible confidential information by stealing the content of the graphic memory with which they were able to reconstruct screenshots of the desktop. As stated by Context, these issues are inherent to the WebGL specification and can't be resolved without major changes in WebGL's architecture. Although there are countermeasures in development, which could resolve these issues, at the moment WebGL allows malicious programs access to the graphics hardware and software. [24], [25]

2.1.5 Status Quo

2.1.6 Evaluation criteria

In the literature, WebGL implementations on mobile devices have been technically evaluated using various methods and range from checking support of official WebGL desktop browser examples to typed array conformance tests to performance tests [3].

2.2 Augmented Reality

Application areas of AR can be as broad as ranging from health care [9] to education [10] to tourism [11]. The topic has gained momentum thanks to the rise in smartphone usage.

2.2.1 History

2.2.2 Categories

AR applications can be categorized in two classes, *AR browsers* and *image recognition-based AR applications* [26].

2.2.3 Native apps and web applications

2.2.4 WebGL usage

2.2.5 Evaluation criteria

User-oriented issues critical for AR evolution and adoption [8]. "AR research still lacks evaluation methods" [27] [12]. Metrics are still very abstract [26]. Obstacles for evaluation the usability of mixed reality systems ("lack of, for example, a common testing platforms and benchmarks")[28]. Thus, low user evaluation rates in AR research, though, many evaluate only early tech demos (cf. [29]). Some evaluation oriented towards user experience [28].

2.2.6 Impact on Human information behavior

Human behavior towards information is influenced by new technologies. recommendation agents - reducing the consumers information overload and search complexity [30]. acquiring product information in in-store settings has often been linked to consumer decision making and information processing [31], [30], [32]. social aspects of mobile image recognition - attaching digital storytelling to physical products - affective influence [33], e.g. trust, engage consumers to communicate and receive information about products [31]

2.3 UI and UX Research

"User experience (UX) is regarded as a holistic concept describing the subjective experience resulting from the interaction with a technological product or service. Both instrumental (e.g. utility, usability, and other pragmatic elements) and non- instrumental (e.g. pleasure, appeal, aesthetics, and other hedonic elements) elements are covered in the UX literature [34]. A recent ISO standard [35] defines UX holistically as a person's perceptions and responses that result from the use or anticipated use of a product, system or service. Recently, pleasurable user experience has become the principal goal in the design of novel interactive systems. Hence, the emphasis on UX has moved the design focus from removing usability and functionality problems or other negative factors to offering possibilities for positive and satisfying experiences that exceed the users expectations."

Research on human-computer interaction (HCI) has been traditionally been rooted in cognitive psychology, engineering, and computer sciences. Besides these fields, research on emotional factors of design is growing [36], which marks a shifts from *usability* to *user experience* analysis.

Academia evaluates user experience oftentimes by looking at the emotional state of the user [37], e.g., by letting users fill out pen-and-paper questionnaires during the course of the usage or by capturing user-made video diaries [38], [39].

3 Implementation

Although data from a gyroscope is accessible with JavaScript and could be therefore used in a WebGL application, we only used data provided by a devices accelerometer, as not all of our test devices had a gyroscope included.

3.1 iOS WebGLViewer

As mentioned in section 2.1.3 there is no official WebGL support in a browser on iOS available. Therefore a simple iOS application, called WebGLViewer, was implemented based on the method and instruction published by Nathan de Vries [19]. WebGLViewer consists of a UIWebView, a reload button and an address bar. The UIWebView was modified to show WebGL content (compare section 2.1.3 and [19]).

If the device is shaken while running WebGLViewer the app shows information about the battery consumption. The current battery level, the battery level when the loading of the current page finished as well as the delta of these two values is shown as percentage. In addition the time the website is shown as well as the battery consumption per minute is shown. Unfortunately the battery level is only updated in 5 % steps. To get a more accurate result the battery information pop-up will automatically be shown if the battery level changes.

The app WebGLViewer was used to view and evaluate WebGL content on iOS.

3.2 Room

4 Evaluation

4.0.1 Battery and CPU usage

To measure the battery usage the WebGLViewer app for iOS was used (see section 3.1). The device was fully charged. The accordant content was opened in the WebGLViewer app. The device then was unplugged and the content reloaded. The time till the battery level reached 90 % was measured by the WebGLViewer. Using Apple Instruments, which is part of Apples Developer Tools, we measured the CPU Activity (Total Activity, Foreground App Activity and Graphics) as well as the relative Energy Usage on a scale from 0 to 20. All test have been performed on an Apple iPhone 4 with iOS 5.1 installed. We compared the battery usage and CPU activity of our room, the Quake 3 WebGL Demo by Brandon Jones [40] and google.com as reference. The results are shown in table 1.

4.1 Frames per second

To test the performance of WebGL on different platforms we measured and compared frames per second (FPS) for different applications on different devices. FPS describes the frequency at which images are generated. Below 12 FPS the human eyes is able to recognize the images as different images, for higher FPS a single image cant be recognized and the images blend together creating motion. So a higher number of FPS creates a more fluid animation and is therefore considered better.

Since the refresh rate of modern flat screens is 60 FPS there is usually no need to calculate more than 60 FPS.

In table 2 we compared the frame rate of the implemented room, the Quake 3 WebGL Demo [40] and a example of a spinning cube, the Spirit Box from webkit.org on different mobile devices and one powerful desktop computer as reference.

5 Conclusion

The conclusion goes here.

Installing WebGLViewer

To install the WebGLViewer using the provided source code on an iOS device a membership in Apple's iOS Developer Program is needed. However compiling and running the WebGLViewer in the iOS Simulator is possible using XCode. As alternative, a registered developer can provide you a binary, which is build for your specific device. We used testflight.com to distribute such binaries. To request a binary you need to create an account on http://bit.ly/zWoZQJ and register your device as soon as you are accepted.

Test Case	Battery and Energy Consumption			CPU Activity		
	Time till 90 %	Battery usage	Relative Energy Usage	Total	Foreground App	Graphics
Our room						
Quake 3	34.67 min	0.289 $\frac{\%}{min}$				
google.com	78.93 min	0.127 $\frac{\%}{min}$				

TABLE 1: Battery consumption of different WebGL applications

Device	Operating System	Browser	Launched	Our room	Quake 3	SpiritBox
Apple iPad 2	iOS 5.1	WebGLViewer	2011	61 FPS	61 FPS	45 FPS
HTC EVO 3D	Android ???	Firefox	2011	8 FPS	12 FPS	N/A [1]
Apple iPhone 4	iOS 5.1	WebGLViewer	2010	40 FPS	30 FPS	43 FPS
Apple iPod Touch (4th Gen.)	iOS 5.1	WebGLViewer	2010	21 FPS	37 FPS	43 FPS
HTC Desire	Android 2.2.2	Opera v12	2010	6 FPS		
HTC Desire	Android 2.2.2	Mozilla	2010	2 FPS	N/A	
Apple iPhone 3GS	iOS 5.1	WebGLViewer	2009			
Reference Computer						

TABLE 2: Frames per Second (FPS) of different WebGL applications on different devices
[1] *The FPS value is constantly alternating between 12 FPS and 209 FPS, making it impossible to determine a realistic value.*

Test	Reference[1]	Desire	EVO 3D	iPhone 4	iPhone 3GS	iPad 2	iPod Touch
3 - Simple animations	59 FPS	FPS	16 FPS	57 FPS	57 FPS	00 FPS	40 FPS
4 - 3D animations	58 FPS	FPS	19 FPS	57 FPS	57 FPS	FPS	57 FPS
5 - Textures	59 FPS	FPS	16 FPS	40 FPS	57 FPS	FPS	40 FPS
6 - Texture filters and keyboard	59 FPS	FPS	FPS	40 FPS	57 FPS	FPS	40 FPS
7 - Basic Lighting	59 FPS	FPS	FPS	57 FPS	57 FPS	FPS	57 FPS
8 - Transparency and blending	59 FPS	FPS	FPS	58 FPS	57 FPS	FPS	58 FPS
9 - Particles	59 FPS	FPS	FPS	39 FPS	31 FPS	FPS	40 FPS
10 - Loading a map	58 FPS	FPS	FPS	57 FPS	57 FPS	FPS	40 FPS
11 - Sphere and rotation	59 FPS	FPS	FPS	40 FPS	57 FPS	FPS	57 FPS
12 - Point lighting	58 FPS	FPS	FPS	40 FPS	57 FPS	FPS	40 FPS
13 - per-fragment lighting	59 FPS	FPS	FPS	36 FPS	40 FPS	FPS	39 FPS
14 - pecular highlights and JSON model	58 FPS	FPS	FPS	35 FPS	39 FPS	FPS	36 FPS
15 - specular maps	58 FPS	FPS	FPS	22 FPS [2]	24 FPS [2]	FPS	22 FPS [2]
16 - Render to texture	58 FPS	FPS	FPS	22 FPS	22 FPS	FPS	22 FPS

TABLE 3: Frames per Second (FPS) using the lessons from the learning WebGL tutorial on different devices
[1] *As reference a 2010 MacBook was used.*
[2] *Rendering results aren't looking as expected*

Appendix B

Appendix two text goes here.

Listing 1: Hello World Code Snippet

```
System.out.println("Hello World!");
```

References

[1] (2012, March). [Online]. Available: https://docs.google.com

[2] (2012, March). [Online]. Available: http://280slides.com/Editor/

[3] D. Golubovic, G. Miljkovic, S. Miucin, Z. Kaprocki, and V. Velisavljev, "Webgl implemenation in webkit based web browser on android platform," in *Telecommunications Forum (TELFOR)*, 2011, pp. 1139–1142.

[4] T. Höllerer and S. Feiner, "Mobile augmented reality," in *Telegeoinformatics: Location-Based Computing and Services*. Taylor and Francis Books Ltd, 2004.

[5] E. Klopfer and K. Squire, "Environmental detectives - the development of an augmented reality platform for environmental simulations," *Educational Technology Research and Development*, vol. 56, no. 2, pp. 203–228, 2007.

[6] J. Vallino, "Interactive augmented reality," Ph.D. dissertation, University of Rochester, 1998.

[7] P. Wellner, W. Mackay, and R. Gold, "Back to the real world," *Communications of the ACM*, vol. 36, no. 7, pp. 24–26, 1993.

[8] T. Olsson and K. Väänänen-Vainio-Mattila, "Expected user experience with mobile augmented reality services. workshop of mobile augmented reality," in *MobileHCI 2011*, 2011.

[9] W. L. D. Lui, D. Browne, L. Kleeman, T. Drummond, and W. H. Li, "Transformative reality: Augmented reality for visual prostheses," in *10th IEEE International Symposium*

on Mixed and Augmented Reality (ISMAR), 2011.

[10] F. Mannuß, J. Rubel, C. Wagner, F. Bingel, and A. Hinkenjann, "Augmenting magnet field lines for school experiments," in *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[11] A. Mulloni, H. Seichter, and D. Schmalstieg, "User experiences with augmented reality aided navigation on phones," in *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.

[12] (2012, March). [Online]. Available: http://www.mobilephonedevelopment.com/archives/1149

[13] (2012, March). [Online]. Available: http://blog.nielsen.com/nielsenwire/?p=29786

[14] S. Wellman. (2012, March). [Online]. Available: http://www.informationweek.com/blog/229215818

[15] (2012, March). [Online]. Available: http://blogs.adobe.com/flashplatform/2011/11/flash-to-focus-on-pc-browsing-and-mobile-apps-adobe-to-more-aggressively-contribute-to-html5.html

[16] (2012, March). [Online]. Available: http://www.khronos.org/news/press/releases/khronos-webgl-initiative-hardware-accelerated-3d-graphics-internet

[17] (2012, March). [Online]. Available: http://www.khronos.org/news/press/releases/khronos-releases-final-webgl-1.0-specification

[18] (2012, March). [Online]. Available: http://blog.vlad1.com/2007/11/26/canvas-3d-gl-power-web-style/

[19] (2011, November). [Online]. Available: http://atnan.com/blog/2011/11/03/enabling-and-using-webgl-on-ios/

[20] (2011, November). [Online]. Available: https://developer.apple.com/library/iad/documentation/UserExperience/Conceptual/iAd_Design_Guide/iAd_Design_Guide.pdf

[21] (2011, June). [Online]. Available: https://www.khronos.org/webgl/public-mailing-list/archives/1106/msg00036.html

[22] (2012). [Online]. Available: https://developer.apple.com/appstore/resources/approval/guidelines.html

[23] (2011, June). [Online]. Available: http://blogs.technet.com/b/srd/archive/2011/06/16/webgl-considered-harmful.aspx

[24] (2011, May). [Online]. Available: http://www.contextis.com/resources/blog/webgl/

[25] (2011, June). [Online]. Available: http://www.contextis.com/resources/blog/webgl2/

[26] T. Olsson and M. Salo, "Online user survey on current mobile augmented reality applications," in *10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, October 2011, pp. 75–84.

[27] M. G. et al., "Experience with an ar evaluation test bed: presence, performance, and physiological measurement," in *ISMAR 2010*, 2010, pp. 127–136.

[28] C. Bach and D. L. Scapin, "Obstacles and perspectives for evaluating mixed reality systems usability," in *Proceedings of the IUI-CADUI Workshop on Exploring the Design and Engineering of Mixed Reality Systems (MIXER)*, 2004.

[29] A. Dünster, R. Grasset, and M. Billinghurst, "A survey of evaluation techniques used in augmented reality studies," in *Proc. ACM SIGGRAPH 2008*, 2008.

[30] T. Kowatch and W. Maass, "In-store consumer behavior: how mobile recommendation agents influence usage intentions, product purchases, and store preferences," *Computers in Human Behavior*, vol. 26, no. 4, pp. 697–704, July 2010.

[31] S. Karpischek and F. Michahelles, "my2cents - digitizing consumer opinions and comments about retail products," in *Proc. of Internet of Things (IOT)*, 2010.

[32] B. Xiao and I. Benbasat, "E-commerce product recommendation engines: use, characteristics, and impact," *Management of Information Systems Quarterly*, vol. 31, no. 1, 2007.

[33] R. Barthel, A. Hudson-Smith, M. Jode, and B. Blundell, "Tales of things. the internet of 'old' things: collecting stories of objects, places and spaces," in *Proc. of the Urban Internet of Things*, 2010.

[34] M. Hassenzahl and N. Tractinsky, "User experience - a research agenda," *Behaviour and Information Technology*, vol. 25, no. 2, pp. 91–97, 2006.

[35] I. organization for standardization, "Iso fdis 9241-2010:2009. ergonomics of human system interaction - part 210: Human-centred design for interactive systems (formerly known as 13407)."

[36] D. A. Norman, "Emotion and design: Attractive things work better," *Interactions*, vol. 9, no. 4, 2002.

[37] V. Roto, "Web browsing on mobile phones - characteristics of user experience," Ph.D. dissertation, Helsinki University of Technology, 2006.

[38] M. Csikszentmihalyi and R. Larson, "Validity and reliability of the experience-sampling method," *Journal of Nervous and Mental Diseases*, vol. 175, no. 9, pp. 526–536, September 1987.

[39] M. Isomursu, K. Kuutti, and S. Väinämö, "Experience clip: method for user participation and evaluation of mobile concepts," in *Proceedings of the eighth conference on Participatory design: Artful integration: interweaving media, materials and practices*, 2004, pp. 83–92.

[40] (2012). [Online]. Available: http://media.tojicode.com/q3bsp/