An HTML Tool for Production of Interactive Stereoscopic Content

Alexey Chistyakov[1](✉) and Jordi Carrabina[1,2]

[1] Aitech S.L., Parc de Recerca UAB, 08193 Bellatera, Barcelona, Spain
alexey.chistyakov@e-campus.uab.cat
http://www.uab.cat, http://www.aitech.es

[2] Escola d'Enginyeria, Universitat Autónoma de Barcelona, CEPHIS,
08193 Bellatera, Barcelona, Spain
jordi.carrabina@uab.cat

Abstract. The benefits provided by the stereoscopic vision have been thoroughly studied for more than a century. The results of this research state that S3D, when properly implemented, enhances efficiency of different applications. At the same time, the market of 3D-enabled technologies is blooming, revealing new devices to the public. 3D cameras, TVs, projectors, monitors, and mobiles are becoming more and more affordable and are not considered as something exotic anymore. Yet this technology is far from being widely implemented. The scarcity of tools for production of such content along with the absence of proper guidelines establishing the rules for S3D depth implementation could be the main reason. With this work we present a tool for generation of interactive stereoscopic HTML compositions. We describe the tool's operation mode and expose results of the performance test.

Keywords: Stereoscopic · 3D · HTML · Human-computer interaction · User interface

1 Introduction

Since the discovery back in XIX century, stereoscopy advanced to a lot of different areas including some that nobody could have ever predicted. The comprehensive analysis of papers and experiments associated with the stereoscopic effect for the last 50 years shows that, in overwhelming majority of cases, usage of stereoscopic displays increases overall performance, if compared to conventional 2D displays, in various tasks [8]. It is known that the stereoscopic effect delivers better and more efficient gaming experience [18,21], and even increases the efficiency of number of medical applications [23]. This could be explained by the way the stereoscopic depth influences our perception of visual information.

As the human-computer interaction is our main research area, we are looking for ways how can HCI benefit from the stereoscopic effect. There are number of studies [1,3,4,24] exploring this influence and resulting to more or less the same encouraging conclusion: stereoscopy has a positive influence user experience

J.M. García-Chamizo et al. (Eds.): UCAmI 2015, LNCS 9454, pp. 449–459, 2015.
DOI: 10.1007/978-3-319-26401-1_42

and overall usability. But, despite relative simplicity of the concept behind the stereoscopic effect, nowadays there are few effective and easy-to-use tools to implement stereoscopic depth into user interfaces (UI).

We see the stereoscopic depth as a tool that allows user interface designers more efficient use of the 3rd dimension in graphical UIs (GUI). These stereoscopic depth planes could present a reliable solution for ambient intelligence system (AmIS) GUIs, which can be heavily cluttered with functionality, controls and, in case of augmented reality UIs, real life objects. Enhanced visual experience and naturalness of a visual image [11] along with a certain level of immersion [14] (if combined with other technological components: e.g. screen resolution, field of view, interaction model) generated by the stereoscopic effect could in turn lead to more natural interaction with the system. A proper approach to this problem could enhance user experience of a system and efficiency of its UI.

In our previous publication [2] we uncovered a 2D-to-3D conversion algorithm (the Algorithm) for HTML-compositions. We managed to update the Algorithm and built a framework, thereby excluding several important restrictions produced by in the initial version of the Algorithm. With this work we try to tackle the problem of introduction of the stereoscopic depth planes into GUIs by developing a tool for production of interactive stereoscopic compositions and validating whether the proposed technology can offer a reliable instrument for that purpose.

2 State of the Art

2.1 Stereoscopic Displays

Stereopsis is a depth perception mechanism that is being thoroughly researched since its discovery more than a century ago [10]. Our binocular vision is what makes this effect possible. Differences and similarities between two images on left and right retinas of two eyes are processed by brain into the information about relative distances and a depth structure of an observed environment. Stereoscopic displays try to follow the same basic principle: provide left and right eye of an observer with two different images, thus the illusion of depth is achieved. There are number of ways of producing such effect [9,16], but the overwhelming majority of them are two-view 3D displays, which all come to the simple idea of splitting so called stereo-pair into two separate images and deliver both images to respective retinas.

Nowadays the most used 3D displays are the *time-sequential two-view displays.* This technology uses interleaving of two images on the same display with a speed fast enough for the viewer not to notice the switching. The images are then filtered by glasses equipped with electronic shutter which delivers left or right image to a respective eye [6]. As of the date of hereby publication this technology is used on most of 3DTVs and 3D-enabled PC monitors available on the market. Thus we chose this type of 3D displays as our main target.

2.2 Platform

We have chosen HTML5 as our main development platform due to a number of features that can help us to build a proper stereo-pair and then provide the observer a way to interact with the output composition.

Though the standard is still in development [7], HTML5 is already in use for several years, supported by many different platforms, and it has one of the widest developer communities. Despite what is hidden under the abbreviation (*Hyper Text Markup Language*), as a standard HTML5 includes three different technologies: HTML, CSS (*Cascading Style Sheet*), and JavaScript. All three combined provide developers with a reliable environment to design and implement cross-platform, screen independent user interfaces.

Deserves mentioning that W3C has the Proposal of Special Extension for Stereoscopic 3D Support [5]. Soonbo H. et al. designed a set of requirements for HTML5 standard aimed specifically at handling stereoscopic content. The proposal contains HTML attributes and CSS properties that provide a possibility to introduce stereoscopic media into a prototype.

2.3 State of the Technology

Most of the tools for production of interactive stereoscopic HTML compositions are associated with WebGL technology [13]. WebGL is a Javascript API for rendering interactive 3D and 2D compositions using only HTML. WebGL elements can be seamlessly implemented into a user interface along with the rest of the HTML. There are several works showing successful implementation of this technology in different areas [12,15,25]. Nevertheless in order to achieve proper results WebGL requires expert knowledge of JavaScript and experience of work with 3D modeling. This problem is alleviated to some extent by a large number of WebGL frameworks, such as ThreeJS [22], even though it still can be considered as the main issue.

3 Method

We plan to use the benefits of all three HTML5 associated technologies. In order to increase the performance of the final application and reduce the time of development we decided to use jQuery [17], the JavaScript framework that was developed to do exactly that. The idea of using the WebGL for this task was declined due to (i) excessive functionality, (ii) complexity of the development process, (iii) lack of support by several targeted platforms, and (iv) overall irrelevancy to the task.

3.1 Conversion Algorithm

Based on the techniques behind the targeted stereoscopic displays and the requirements of HTML5 standard we designed the Algorithm of conversion of an input markup into a stereo-pair that preserves all the interactive features presented before the conversion. The Algorithm's operation mode can be divided into 4 main stages: (S1) cloning of the existing content and creating a side-by-side composition; (S2) detecting stereoscopic content, such as images, videos etc.; (S3) mirroring the interactions from one side to the other; and (S4) distributing user predefined markup elements along the depth axis (z-axis) the way the illusion of depth is achieved. At the same time, several challenges (see Table 1) had to be alleviated in order to achieve the desired result were defined.

Table 1. Main challenges to overcome during the 2D to 3D conversion process

HTML	(1) The specific of HTML as a language allows multiple variations of the code producing the same layout, all of this variations should be taken into account during the conversion process;
	(2) The output HTML should correspond to the W3C requirements all the violations should be predicted and eliminated
	(3) The invalid input HTML code that could sabotage the output result should be predicted and taken into account;
	(4) The tool needs to recognize the proposal for W3C requirements [5] aimed at handling the stereoscopic media content through respective *@media* queries.
CSS	(1) Multiple variations of style rules is allowed by the standard, such as (i) inline styling, (ii) styling through HTML's *styles* tag, (iii) styles inside inked *.css* file, (iv) using *@import* inside linked *.css* files. All the variations should be taken into account and processed;
	(2) Interaction related CSS pseudo classes, e.g. *:hover*, *:focus*, should be taken into account during the interaction mirroring stage of the conversion Algorithm;
	(3) During the shifting of the specified elements along the z-axis the rest of the HTML layout should not be disturbed.
JavaScript	(1) The tool should be easy to include on the page and require as minimum user input as possible;
	(2) The framework has to provide access to both original and clone elements at the same time during the programming of interactions with the output stereo-pair.

3.2 The Implementation

The final version of the tool (3DSjQ) exploits the Algorithm and successfully alleviates all the aforementioned challenges. During the operation, 3DSjQ recommended itself as a stable working easy-to-use tool. Due to the HTML in its core, the output composition can be viewed on 3D enabled projectors, TVs, PC monitors, tablets, smartphones, etc. The tool allows front-end developers to design all type of interactions supported on a client side and work with any type of user input devices, including basic ones, such as mouse, keyboard, touch

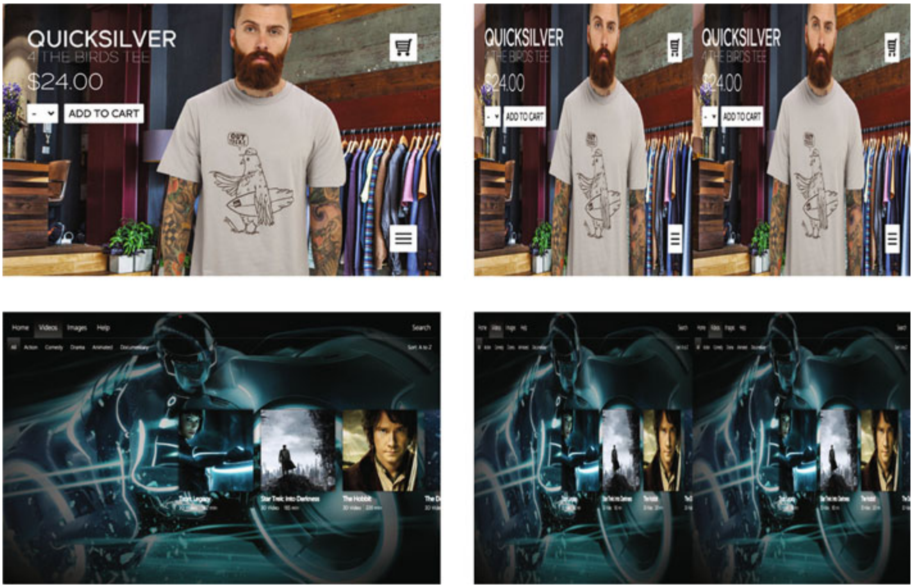


Fig. 1. Stereoscopic interfaces for the e-commerce application (top) and the stereoscopic video browser (bottom) created with 3DSjQ. User interface before conversion (left) and output stereo-pair after conversion (right).

screens, and more sophisticated like MS Kinect, Leap, and any other equipped with a corresponding API.

Since the first version presented earlier [2] 3DSjQ went through a number of optimization cycles and drastically advanced its functionality, features and stability. For comparison of features presented in both releases please refer to the Table 2.

Despite the great improvement in stability, the number of features and the fact that the latest version of the framework can predict invalid HTML code, 3DSjQ still has several restrictions to composition design and further HTML implementation caused by browser rendering issues and specifics of the Algorithm. For example, (i) it is strongly advised to maintain validity of the input code and its correspondence to the W3C requirements; (ii) all the interactions with the output stereo-pair should be specified after 3DSjQ was executed since all the javascript executed prior the tool was triggered could cause interaction fails; (iii) due to browser rendering issues some graphical stylings can look different from the input composition, thus visual properties of some elements need to be corrected correspondingly;

3.3 Operation

We designed 3DSjQ the way it requires only the basic knowledge of HTML and related technologies. The procedure of enabling the framework on the page similar to any other jQuery plugin: the *3DSjQ.js* file should be specified in a header

Table 2. 3DSjQ features

Feature	v.1.0
Files included in the package	1
Lines of code	1387
Size of the developer version, (kb)	29.8
Ability to set parameters of the stereoscopic environment[a]	Yes
Ability to convert already marked up prototypes	Yes
Framework of functions to manipulate the output	Yes
Prevention of code fails	Yes
Offline work	Yes[b]
Support of W3C requirements for stereo web [5]	Yes
- Stereoscopic videos support	Yes
- Stereoscopic images support	Yes
*@media queries*ic *@media queries*	Yes

[a] *stereo−pair arrangement* (*top-bottom, left-right*), *visual cues, depth budget, etc.*

[b] *most of the browsers*

of an HTML page to convert. Then, in order to start the conversion, one should launch the initiating function *go3DS elements, params*), where *elements* is an array of elements to shift and *params* array of parameters of stereoscopic depth.

There are two ways of shifting a certain element away from a screen plane: (i) it can be specified within the parameters of the initiating function (see Listing 1.1), this way the element is going to be shifted right after the conversion has been executed, or (ii) with the *.shift*() function at any time after the conversion has been carried out (see Listing 1.2). This function, along with others completes the framework, which gives a possibility to distribute elements along z-axis.

```
...
   <script type="text/javascript" src="jQuery.js"></script>
   <script type="text/javascript" src="3dsjq.js"></script>
   <script type="text/javascript">
      $(window).on({ load: function(){
         go3DS({ "#foo" : 3, "#foo:hover" : 5 });
      }});
   </script>
   </head>
...
```

Listing 1.1. Basic initiation of the 3DSjQ, during which HTML element with *id="foo"* will be put 3 levels above the screen plane in a standby mode and 5 levels above when a *:hover* event is triggered.

```
$("#foo").shift(5);
$("#foo").parent().shift(3);
$("#foo").on({
    mouseenter: function(){
        $(this).parent().shift(3);
    }
});
```

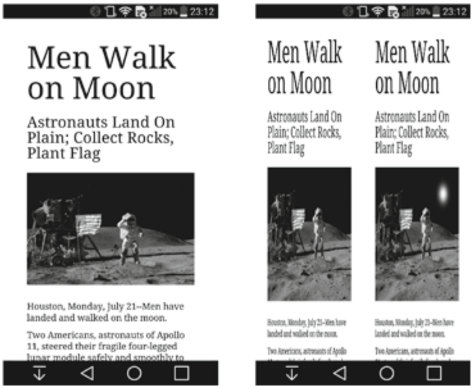
Listing 1.2. Possible ways to shift elements away from a screen plane. From top to bottom: (i) shifting of the *#foo* element to a 5th level, (ii) shifting of a parent of the *#foo* to a 3rd level, (iii) shifting the parent the 3rd level when the *mouseenter* event was triggered.

4 Performance Evaluation

Each browser has specific properties and rendering capabilities which can affect the output markup. Moreover, operating system that is hosting the browser can also influence the rendering behavior. Thus, in order to evaluate the obtained tool in terms of performance and platform compatibility, we tested it on the most popular browsers (desktop and mobile) under the most popular operating systems [19,26] (see Table 3). We included in the research the following browsers according to its current market share: *Google Chrome* - 43.16 %, *Safari* - 13.8 %, *Internet Explorer* - 13.09 %, *Android Browser* - 6.99 %, *Opera* - 3.79 %. Other browsers were also tested, but they are not included into hereby research due to their low market share. As of operating systems, we included *Windows 7* - 33.9 %, *Android* - 20.67 %, *iOS* - 11.51 %, *Windows 8.1* - 9.26 %, *OS X* - 5.69 %.

The displays that have been tested include: Samsung 3DTV with screen size of 1920x1080px, Samsung UD590 with 2560x1440px, Apple iPhone 5S - 1136 x 640px, LG G3 - 2560x1440px. It should be noted that a screen size, when requested by javascript and returned by a browser, is different depending on the display resolution, for example Apple iPhone's screen size is recognized as 320x568px. Thus, the aforementioned devices available at our disposal can be considered as a reference group for the currently the most used screen sizes [20]. We also added Tizen OS and PS4 System Software to the test. Though browsers hosted by those OS' can not be considered as popular, yet both platforms can benefit from stereo content due to their entertainment oriented purpose.

For the test purpose we created several responsive prototypes exploiting stereoscopic features provided by the 3DSjQ. The HTML markup of all the prototypes is done the way it can adapt its appearance to a current display screen size (also known as responsive). The prototype was developed on a desktop PC under Mac OS X 10.10.3 and tested on Google Chrome browser along the process, thus this browser-OS combination was used as a main reference.



Men Walk on Moon

Astronauts Land On Plain; Collect Rocks, Plant Flag

Houston, Monday, July 21–Men have landed and walked on the moon.

Two Americans, astronauts of Apollo 11, steered their fragile four-legged lunar module safely and smoothly to

Fig. 2. The initial HTML prototype (to the left) and the output stereo-pair (to the right) containing the stereoscopic image (courtesy of the NASA) as seen on Chrome Browser under Android OS

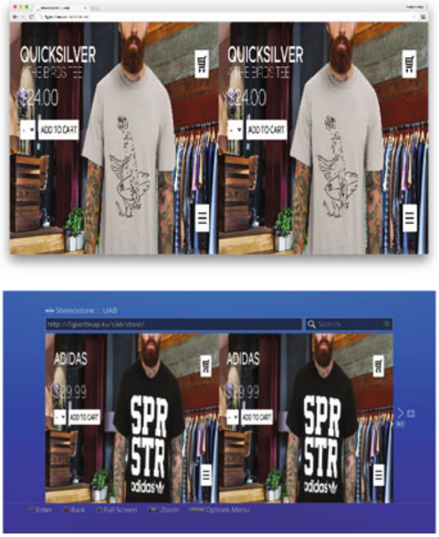


Fig. 3. Output stereo-pair as seen on Google Chrome under Mac OS X (top) and PS4 browser (bottom)

5 Results

As seen in Table 3 the framework performs flawlessly in terms of support of its features when tested under different operating systems. The final versions of the stereoscopic prototypes were compatible with almost all of the targeted platforms. Overall, the 3DSjQ was able to run on all the test devices with most of the visual features displayed correctly. Some browsers were unable to correctly render

Table 3. 3DSjQ browsers compatibility

	Chrome	Safari	IE	Android Browser	Opera	Other
Windows 7	✓	✓	x	–	✓	–
Android	✓	–	–	x	✓	–
Windows 8.1	✓	✓	✓	–	✓	–
iOS	✓	✓	–	–	✓	–
OS X	✓	✓	–	–	✓	–
Tizen	–	–	–	–	–	✓
PS4	–	–	–	–	–	✓

– - not applicable , ✓ - supported, x - not supported

the output composition due to incompatibility of the input HTML, though the framework performed as expected. The obtained results will be taken into consideration and will be useful for testing framework's fallback strategies.

6 Conclusion

With this work we proposed the tool for production of passive and interactive stereoscopic compositions which can be considered as a way to alleviate this problem and represents an effective instrument for developing S3D user interfaces for human-computer interaction research purposes. The framework was in a state of development for several years, and, as shown in this work, performs really well under all the most popular browser versions and operating systems. It allows developers to design complex interactions with a number of user input devices, such as keyboard, mouse, touch screens and many others that have a proper APIs. 3DSjQ can be used for production of stereoscopic user interfaces from scratch and as a conversion tool for already implemented HTML, which makes it one of a kind tool for front-end development of S3D applications.

7 Future Work and Discussion

Despite the outstanding technical performance, there is still work to be done to improve stability and overall functionality of the framework. This can include: (i) further update of the functionality, (ii) closer work with the stereoscopic web standards in order to support its features and contribute to the standard's development; (iii) find fallback strategies for browsers that can not support all the features provided by the HTML5; (iv) provide support for autostereoscopic displays. Taking into account the interaction related issues mentioned before, it is important to make a proper user evaluation of compositions built with the presented tool in order to substantiate the described approach.

We plan, and encourage other HCI researchers, to use this method to develop S3D stimuli for further research work on human interaction with S3D displays, which tend to become ubiquitous. It could help to establish guidelines and standards for efficient usage of the stereoscopic depth, test different S3D interaction paradigms, measure level of immersivenes and similar issues. In other words, find ways to design better UX and more efficient UIs.

Acknowledgments. Funded by the Catalan Government scholarship 2013-DI-030

References

1. Broy, N., Schneegass, S., Guo, M., Alt, F., Schmidt, A.: Evaluating stereoscopic 3D for automotive user interfaces in a real-world driving study. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA 2015, pp. 1717–1722. ACM Press, New York, April 2015. http://dl.acm.org/citation.cfm?id=2702613.2732902
2. Chistyakov, A., González-Zúñiga, D., Carrabina, J.: Bringing the web closer: stereoscopic 3D web conversion. In: Collazos, C., Liborio, A., Rusu, C. (eds.) CLIHC 2013. LNCS, vol. 8278, pp. 22–25. Springer, Heidelberg (2013)
3. González-Zúñiga, D., Chistyakov, A., Orero, P., Carrabina, J.: Breaking the pattern: study on stereoscopic web perception. In: Urzaiz, G., Ochoa, S.F., Bravo, J., Chen, L.L., Oliveira, J. (eds.) UCAmI 2013. LNCS, vol. 8276, pp. 26–33. Springer, Heidelberg (2013)
4. Häkkilä, J., Posti, M., Koskenranta, O., Ventä-Olkkonen, L.: Design and evaluation of mobile phonebook application with stereoscopic 3D user interface. In: CHI 2013 Extended Abstracts on Human Factors in Computing Systems on - CHI EA 2013, p. 1389 (2013), http://dl.acm.org/citation.cfm?doid=2468356.2468604
5. Han, S., Lee, D.Y.: Extensions for Stereoscopic 3D support (2012). http://www.w3.org/2011/webtv/3dweb/3dweb_proposal_121130.html. Accessed 6 Jun 2015
6. Hartmann, W.J., Hikspoors, H.M.J.: Three-dimensional TV with cordless FLC spectacles. Inf. Disp. **3**(9), 15–17 (1987). http://dl.acm.org/citation.cfm?id=33861.33864
7. Hickson, I.: A vocabulary and associated APIs for HTML and XHTML (2012). http://www.w3.org/TR/html5/. Accessed 5 Jun 2015
8. Holliman, N.: 3D display systems. In: Handbook of Opto-electronics (2005). ISBN 0-7503-0646-7
9. Holliman, N.S., Dodgson, N.A., Favalora, G.E., Pockett, L.: Three-dimensional displays: a review and applications analysis. IEEE Trans. Broadcast. **57**(2), 362–371 (2011)
10. Howard, I.P.: Seeing in depth, Vol. 1: Basic mechanisms. University of Toronto Press (2002)
11. IJsselsteijn, W., de Ridder, H., Hamberg, R., Bouwhuis, D., Freeman, J.: Perceived depth and the feeling of presence in 3DTV. Displays **18**(4), 207–214 (1998)
12. Johnston, S., Renambot, L., Sauter, D.: Employing WebGL to develop interactive stereoscopic 3D content for use in biomedical visualization. In: Dolinsky, M., McDowall, I.E. (eds.) The Engineering Reality of Virtual Reality 2013, Proceedings of the SPIE, vol. 8649, p. 864–905, March 2013. http://adsabs.harvard.edu/abs/2013SPIE.8649E.05J
13. Khronos WebGL Working Group: WebGL - OpenGL ES 2.0 for the Web (2015). https://www.khronos.org/webgl/. Accessed 10 Jun 2015

14. McMahan, R.P., Gorton, D., Gresock, J., McConnell, W., Bowman, D.A.: Separating the effects of level of immersion and 3D interaction techniques. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology - VRST 2006, p. 108. ACM Press, New York, November 2006. http://dl.acm.org/citation.cfm?id=1180495.1180518
15. Nocent, O., Piotin, S., Benassarou, A., Jaisson, M., Lucas, L.: 3D displays and tracking devices for your browser: A plugin-free approach relying on web standards. In: Proceedings of the 2012 International Conference on 3D Imaging - IC3D 2012, pp. 1–8, December 2012. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6615141
16. Pastoor, S., Wöpking, M.: 3-D displays: a review of current technologies. Displays **17**(2), 100–110 (1997)
17. Resig, J.: jQuery (2006). https://jquery.com/. Accessed 5 June 2015
18. Schild, J., Bölicke, L., LaViola Jr., J.J., Masuch, M.: Creating and analyzing stereoscopic 3D graphical user interfaces in digital games. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI 2013, p. 169. ACM Press, New York, April 2013. http://dl.acm.org/citation.cfm?id=2470654.2470678
19. StatCounter: StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share (2015). http://gs.statcounter.com/#all-browser-ww-monthly-201502-201502-bar. Accessed 8 Jun 2015
20. StatCounter: StatCounter Global Stats - Browser, OS, Search Engine including Mobile Usage Share (2015). http://gs.statcounter.com/#desktop+mobile+tablet-resolution-ww-monthly-201502-201502-bar. Accessed 8 Jun 2015
21. Takatalo, J., Kawai, T., Kaistinen, J., Nyman, G., Häkkinen, J.: User experience in 3D stereoscopic games. Media Psychol. **14**(4), 387–414 (2011)
22. three.js: three.js - Javascript 3D library (2015). http://threejs.org/. Accessed 8 Jun 2015
23. Van Beurden, M., IJsselsteijn, W., Juola, J.: Effectiveness of stereoscopic displays in medicine: a review. 3D Rese. **3**(1), 1–13 (2012)
24. Ventä-Olkkonen, L., Posti, M., Häkkilä, J.: How to Use 3D in Stereoscopic Mobile User Interfaces. In: Proceedings of International Conference on Making Sense of Converging Media - AcademicMindTrek 2013, pp. 39–42 (2013). http://dl.acm.org/citation.cfm?id=2523447
25. Wang, W., Dong, S., Wang, R., Cheng, Q., Zhang, J., Liu, Z.: Stereoscopic 3D Web: from idea to implementation. In: 2014 International Conference on Information Science, Electronics and Electrical Engineering, vol. 3, pp. 1440–1444. IEEE, April 2014. http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6946158
26. Zachte, E.: Wikimedia Traffic Analysis Report - Browsers e.a (2015). https://stats.wikimedia.org/archive/squid_reports/2015-02/SquidReportClients.htm. Accessed 8 Jun 2015