



Vr & Web Gui Shell: Interactive Web-System For Virtual Reality

Darina Vladimirovna Vorobyeva¹, Alexey Dmitrievich Leukhin¹,

Vlada Vladimirovna Kugurakova¹

1 Kazan Federal University

2 Kazan Federal University

3 Kazan Federal University

ABSTRACT

In this article, we provide the reader with an idea of realisation an interactive web system for virtual reality (VR). First, we discuss the role of web system for VR, and its relevance and importance. Then, we provide a brief overview of the existing technologies, which allows to implement this idea at the software level. Particularly, we review the possibilities of implementing the functions of the system, which will be gathered in one place: display of standard video and video 360°, with dynamic playlist and descriptions windows; online maps, where the mode of «street view» will be in the focus of attention; output web pages, based on automatic transformation or ready-made «VR mode» of WebVR technology; and the adaptation of the browser to VR, which should solve one of the important problems – user-friendly navigation and ensuring high performance because VR in web has a lot of extra interlayers, instructions translation and etc., and this all decreases the performance. Thus, we are attempting to swap them and make a web-system in the virtual reality system. At the same time, we present our concepts, which clearly demonstrate our idea. Finally, we offer some possible ways of our web-system development in the future.

Keywords: virtual reality, web system, interactive, concepts, analysis



1. INTRODUCTION

Nowadays virtual reality and augmented reality are coming to the fore and becoming very popular not only in computer games. Interest in technology has grown, when researchers developed devices for their integration with browsers. WebVR technologies and tools – which are embed in web-browsers and allow to model 3D objects without using own hardware and software platforms – are developing very fast.

For instance, in the browser WebVR can interact with modern terminals such as Oculus Rift, Google Cardboard, HTC Vive, etc (WebVR Rocks. 2017). In fact, WebVR offers common standards, which are followed by the creators of browsers, developers of hardware solutions and content producers. It allows to avoid a huge amount of devices and incompatible solutions, users do not have to choose which platform should they prefer, and content developers have common set of rules that can be easily followed (Mozilla VR team. 2017).

However, using vr-in-web technology has some problems, which require early solution.

2. RELATED WORK

During the work there was a comparative analysis of existing similar projects based on the information provided on official websites and review materials (Samsung Internet for Gear VR. Retrieved from www.oculus.com/experiences/gear-vr/849609821813454, Best VR web browser: how to browse the internet in virtual reality. Retrieved from www.techradar.com/news/internet/web/best-vr-web-browser-how-to-browse-the-internet-in-virtual-reality-1317762). The result is demonstrated in Table 1.

Table 1. Results of comparative analysis

Feature / Device	Samsung Gear VR browser	MozVR	JanusVR	VR&web GUI Shell
Video Stream 3D and 360°	+	-	-	+(Vulkan + WebVR)
Google Maps (interactive street view mode)	-	-	-	+(Vulkan + WebVR)



Applications	+	+	-	?
Support mobile devices	+	+	-	-
Web page 3D	± (some pages)	-	± (as a game)	+ (WebVR + Vulkan)
Workplace environment (tabs, windows)	± (only tabs)	± (only tabs)	± (as a game)	+ (Vulkan)
Cross platform	- (Android)	-	+	+ (Vulkan)
Cross browser	-	?	?	+ (Vulkan)
HTML5 support	+	+	+	+ (WebVR)

The solutions that we offer should eliminate defects of other projects to be significant and solve specific problems of VR.

3. THE PROBLEM

The problem of vr-in-web technology is low speed, because the browser has a lot of extra interlayers, instructions translation and etc., and this all decreases the performance. Moreover, browsers tend to be less optimized for hardware and software features of specific devices. In addition, the technology is not well-developed yet, some functions are not implemented, sometimes there are mistakes and incompatibilities. Many things are still in beta status, even browsers (Google Developers. 2017.).

The transparency of the transition between the real world and the virtual is required, it is necessary to discard the frames/limitations imposed by browsers, and the conventionality of the solutions offered by browser frameworks for visualization in the presentation of a visual series.



Finally, vr-in-web technology has the same problem, as VR entirely has – the best way to use it is not found yet. It is not the matter of the technology, but the matter of user experience (UX) and user interface (UI).

4. OUR IDEA

Proceeding from the listed problems, we propose to swap VR and web and make web-in-vr. Based on the proposed paradigm, a prototype web-in-vr is being developed with different interface solutions, that uses vr-in-web features in part. To estimate it, it is necessary also to develop new metrics for quantify usability – including learningability, the metric that was proposed in the article at the DESE2017 conference (Manakhov et al, 2017), measuring the effect of using virtual laboratories (Abramov et al, 2017). So we present an idea and concepts for web browser (or another words «web GUI shell») for VR that gives the user basic functionalities which extended for VR without having to leave the virtual space.

The idea of creating a browser for VR has been already voiced in the world. There are some similar developments (Barsoum and Kuester,2005), but sometimes their functionality is severely limited, usually exclusive for a headset or presented in a game form (Cervantes,2016). The main purpose of our web-in-vr system is to enjoy 3D and 360-degree media content, walk the streets of another country by Google maps VR and increase quality and learn productivity of found information by multimedia features. Also make it cross platform.

We propose to consider some separate basic elements and the concepts attached to them.

5. SOLUTION AND CONCEPTS

To solve performance problem it will be more appropriate to use WebVR not in all functions of browsers, but only for presentation of web-content and video. Rendering of browser elements and maps will be partially formed from other technologies.

Our associative links when using web-in-vr in conditions of new UX solutions should get a natural representation.

«Recent technologies, such as CSS grid, WebGL or Web Components clearly showed that we had already passed the period when it was necessary to work with incorrect, incompatible implementations in different browsers. Most of the browser developers



have reached a mutual understanding, and knowing that such technologies as Web Assembly will take place soon, in future we will have more opportunities to create different versions of virtual reality, working in all browsers (Splitt,2016)»

– *Martin Splitt.*

However, even now web developers can easily create objects in space due to a well-developed and rich WebVR specification.

Videos

In March 2015, the largest video hosting Youtube presented support for a video with a 360-degree view, that allows to watch clips from different angles. This is a completely new level of human immersion in video content.

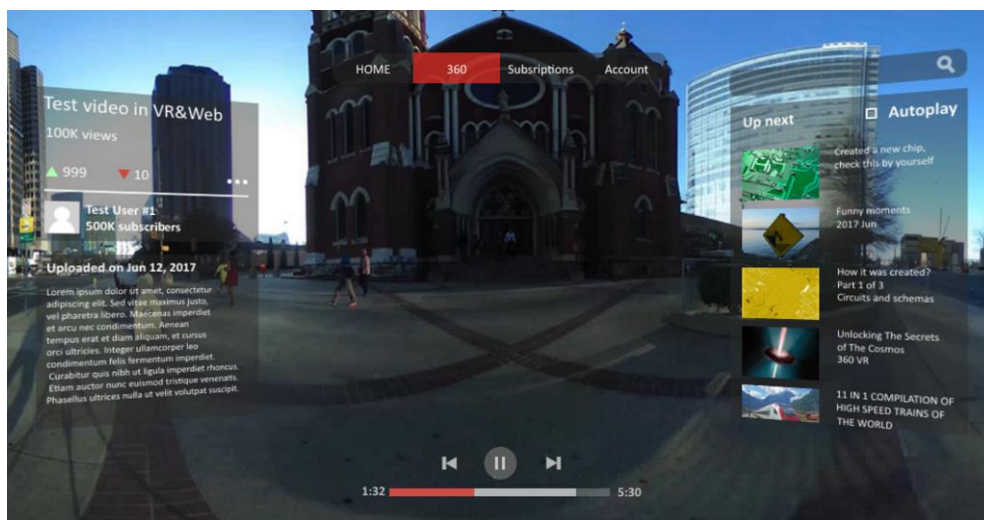


Figure 1. Concept of video display with functional elements and YouTube style.

Using API: Vulkan + WebVR (A-Frame).

However, using video with high resolution reduces the feeling of «immersion», because the video can be played on light devices with a delay. It must be avoided in web-in-vr system, and this problem was solved by the group Feng Qian, who demonstrate in 2016 their approach (Qian, 2017), which can reduce bandwidth consumption based on a trace-driven simulation.

Features. It is necessary to support not only watching videos, but also the functionality of video hosting (for example, YouTube). For this it is essential to develop the mechanism of user interaction with hosting elements (video rewind, playlist, description to video, search and interaction with account). The concept Fig.1 represents a possible



version of display the elements before the video start with their further concealment at the beginning of the watching or display by the call of a button on the manipulator.

Manipulators as the most convenient control mechanism for video (Pakkanen et al, 2017) are needed for interacting with all elements of the system.

Maps

Walking around the world is one of the biggest possibilities of using VR technology. The application Google Street View is filled with 360-degree view images, hosted by Google and other users, but this is only for phones, only for interesting (but not for all) places on the planet and without the opportunity to go through all the streets as in normal panorama mode in maps.

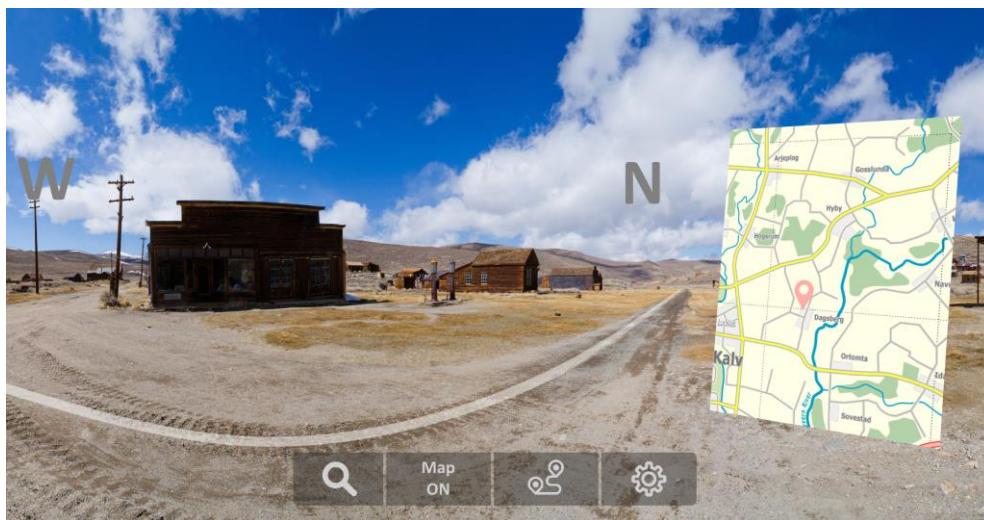


Figure 2. Concept of interactive street view with small global map and menu at the bottom.

Using API: Vulkan + WebVR (only for catching head movement and manipulators).

As appropriate API for our specific task is absent (on request on the Internet was not found), it is necessary to implement it basing on available technologies and own code.

We describe the structure on high level. Vulkan is taken as a basis. The transfer of head movement (overview) occurs through WebVR, which provides this opportunity.

Manipulators move the user on the map. It does not seem like something new, but it is a possibility to walk on the all streets (not only in selected areas as in the application StreetView on the phone) with new additional features.



Features. The concept Fig.2 demonstrates additional elements extending the main content provided by Google Maps – it is the mini map on the right (it can be grabbed and moved by the manipulator), which shows the location on the global map. Ability to display (overlapping) the sides of the world on the panorama and also building a route. In plans to combine the capabilities of the Google application with web-in-vr.

Websites

The main purpose of demonstration of web-sites in VR is to break the old browser «window» metaphor. Thus, the potential of VR will be fully revealed, cause why do you have to use 2D sites in the headset if it can be done with the same success and greater speed on the monitor with a keyboard and mouse?

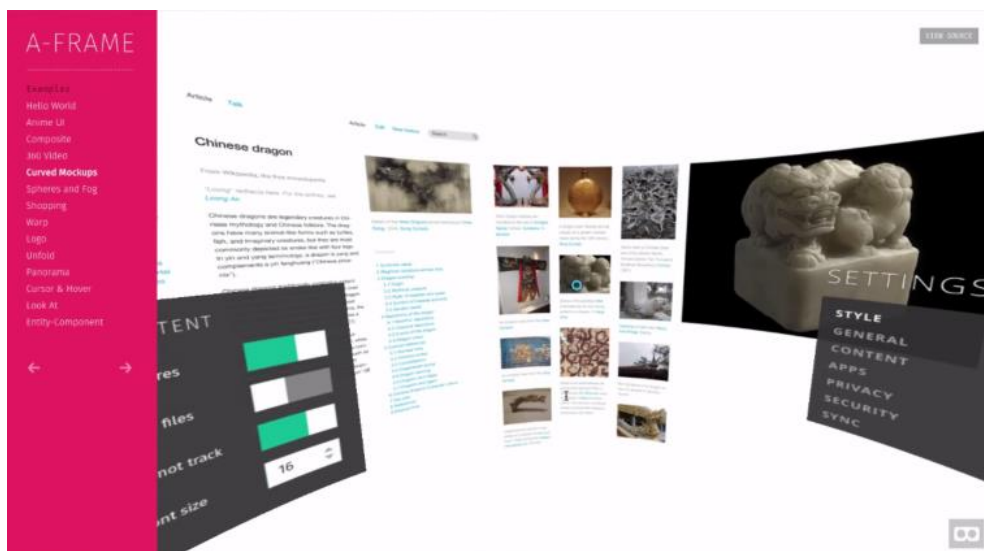


Figure 3. Screenshot of A-frame Wikipedia website sample (Zheng,2017).

Using API: Vulkan + WebVR (A-Frame).

It is possible to create VR-world with minimal effort by using A-Frame. A-Frame is a HTML-framework for creating web-applications and sites for virtual reality. In the browser it looks like a web-page, that represents 3D-image with ability to navigate and interact with objects. The images are rendered by WebGL.

The main purpose of A-Frame is to provide a simple tool, that helps to create 3D-world in a way similar to HTML markup. The framework is a high-level add-in based on Three.js (5).



Features. The concept Fig.3 gives an example of displaying a web-site in VR mode, where there is a possibility of flexible settings. Multimedia material also can be displayed in panorama or in 360, if it is provided by the site. Here, the web-in-vr system uses the code implemented on the page for VR and as vr-in-web displays objects in the space.

Web Gui Shell

The idea of «Web GUI shell» is in using browser engine wrapped in a new shell for VR. It is necessary to destroy the old browser «window» metaphor and show the infinity of VR. It is essential to use powerful tools.



Figure 4. Concept of browser functions which interpreted as separate elements.

Using API: Vulkan.

To visualize «browser» in VR were decided to use Vulkan. It is a low-overhead, cross-platform 3D graphics and compute. Vulkan is intended to provide a variety of advantages over other APIs as well as its spiritual predecessor, OpenGL. Vulkan offers lower overhead, more direct control over the GPU, and lower CPU usage (Khronos.org. 2015).

Features. The most important thing is to present all the main elements of the browser (tabs, history, downloads) in a user-friendly way in the form of separate objects Fig.4. Clicking on them their content reveals in the space and the user can interact with the



target elements more easily as the other objects will be hidden and will not interfere with the viewing.

Thus, in conjunction with the previously described functional (video, maps, websites) a certain bridge is built between the user and the Internet, when the information is presented in a new, somewhat unusual, but exciting form. So, a person can differently perceive and absorb information.

6. EXPERIMENTAL RESULTS

To choose the technology it is necessary to consider the important characteristics such as performance, cross platform, usability and full API user manual. For our purposes most suited Vulkan and OpenGL and we have tested them. Library files and C++ commands for testing were taken from the GitHub repository (GitHub repository: Comparison of OpenGL and Vulkan API in terms of performance. Retrieved from github.com/RippeR37/GL_vs_VK). Table 2 gives brief descriptions of these tests.

Table 2. Test descriptions. Test №2 and №3 doesn't have multithreaded version with OpenGL API as they don't have easily scalable CPU-bound work that could benefit from dispatching to multiple threads.

Test description	Multithreaded OpenGL	Multithreaded Vulkan
Static scene. Resolves around single static scene with variable number of rendered objects which quality can be chosen (each is a sphere with specific amount of vertices)	yes	yes
Terrain with dynamic LoD. A terrain is rendered with active dynamic level-of-detail changes. It loads 1024x1024 heightmap and pre-computes any necessary data	no *	yes
Shadow mapping. Rendering a "checkboard" floor with differently-colored cubes and one high-res sphere and many cubes in different positions	no *	yes



We have tested OpenGL and Vulkan without multithreading to put them under the same conditions (Vulkan with parallelization will be more faster than OpenGL). The results are presented in Table 3.

Table 3. Result of testing.

Video card	Processor	Test	OpenGL 4.5	Vulkan 1.0.42	Diff
Nvidia 720M	Intel Core i3- 3217U 1.80GHz (2 cores)	Static scene	3.13	5.21	39.9%
		Terrain with dynamic LoD	20.37	27.10	24.8%
		Shadow mapping	58.72	46.13	21.4%
Nvidia 840M	Intel Core i5- 4210U 1.70GHz (2 cores)	Static scene	3.95	6.62	40.3%
		Terrain with dynamic LoD	27.26	33.14	17.7%
		Shadow mapping	73.52	44.16	39.9%
Nvidia 970M	Intel Core i7- 4710MQ 2.5 GHz (4 cores)	Static scene	5.62	9.09	38.2%
		Terrain with dynamic LoD	45.11	54.94	17.9%
		Shadow mapping	146.84	258.97	43.3%
Nvidia GTX 780 Ti *	Intel Core i7- 7700K 4.50GHz (8 Cores)	Static scene	7.32	12.67	42.2%
		Terrain with dynamic LoD	51.48	63.68	19.1%
		Shadow mapping	178.36	307.98	42.0%
Nvidia GTX	Intel Core i7-	Static scene	7.43	12.66	41.3%



1050 *	7700K 4.50GHz (8 Cores)	Terrain with dynamic LoD	51.40	63.92	19.5%
		Shadow mapping	178.49	273.16	34.6%

* this result was taken from openbenchmarking.org. The full list of another video cards can be found in the Michael Larabel's article (2017) (Larabel. 2017).

Based on performance test results Vulkan is optimal. In addition, it has a number of other advantages:

1. it is the first low-level API of the new generation, which is cross-platform;
2. Vulkan has low latency and high efficiency, so developers can add more details and special effects while maintaining excellent performance;
3. it differs from others in the efficient support of multithreading and uniform load on all cores.

At this stage of the project, game engines with more advanced functionality such as UE4 and Unity3D were not considered, because there is no way to work with information directly, at a low level, as applications built with such engines will in any case be inferior in performance, because the processing of information will take place in several stages of compilation and translation between levels of the game engine.

7. CONCLUSION

Web-in-VR system solves some local problems of vr-in-web technology, but no answer for a global problem: most web developers don't have the skills to create VR. The expertise needed to build a conventional website is totally different from that needed to building an interactive environment. The next global problem is adapting to VR design where terminology is still being defined.

No doubts, that it is more easily and efficiently to make a small site on HTML/CSS and show it to a billion potential users on the displays of smartphones, but, on the other hand, these users have already had the access to a huge array of products. Therefore, in order to be outstanding, it is necessary to look for the new ways of expression and interaction patterns.

As pioneers within this industry, the possibilities are endless.

8. ACKNOWLEDGEMENT



This work was funded by the subsidy of the Russian Government to support the Program of competitive growth of Kazan Federal University among world class academic centers and universities.

9. REFERENCES

- Abramov Vitaly, Kugurakova Vlada, Rizvanov Albert, Abramskiy Mikhail, Manakhov Nadir, Evstafiev Mikhail, and Ivanov Denis. 2017. Virtual Biotechnological Lab Development. *BioNanoScience* 7, 2 (01 Jun 2017), 363–365. DOI: 10.1007/s12668-016-0368-9
- Best VR web browser: how to browse the internet in virtual reality. Retrieved from www.techradar.com/news/internet/web/best-vr-web-browser-how-to-browse-the-internet-in-virtual-reality-1317762
- Cervantes Edgar. 2016. Best VR browsers – immerse yourself into the web. (2016). <http://vrsource.com/best-vr-browsers-immersed-web-5538>
- GitHub repository: Comparison of OpenGL and Vulkan API in terms of performance. Retrieved from github.com/RippeR37/GL_vs_VK
- Barsoum Emad and Kuester Falko. 2005. WebVR: an interactive web browser for virtual environments. *Proc. SPIE* 5664 (2005), 540–547. DOI: 10.1117/12.582624
- Google Developers. 2017. Web Fundamentals: WebVR. (2017). <https://developers.google.com/web/fundamentals/vr>
- Hedelin Hugo. 2017. Design and evaluation of a user interface for a WebVR TV platform developed with A-Frame. Master's thesis. Linköping University, Department of Computer and Information Science.
- Khronos.org. 2015. Vulkan - Graphics and compute belong together. (2015). <https://www.khronos.org/vulkan/>
- Larabel Michael. 2017. GL_vs_VK: A Micro-Benchmark Looking At The Overhead Of OpenGL vs. Vulkan APIs. (2017). Retrieved from <http://www.phoronix.com/scan.php?page=article&item=gl-vs-vk>
- Mozilla VR team. 2017. Introduction – A-Frame. (2017). Retrieved from <https://aframe.io/docs/0.6.0/introduction>



- Manakhov Nadir, Kugurakova Vlada, Abramskiy Mikhail, Abramov Vitaly, and Maslaviev Azat. 2017. Visual editor of scenarios for virtual laboratories (10th International Conference on the Developments on eSystems Engineering). Paris, France.
- Pakkanen T., J. Hakulinen, T. Jokela, I. Rakkolainen, J. Kangas, P. Piippo, R. Raisamo, and M. Salmimaa. 2017. Interaction with WebVR 360; video player: Comparing three interaction paradigms. (March 2017), 279–280. DOI: 10.1109/VR.2017.7892285
- Samsung Internet for Gear VR. Retrieved from www.oculus.com/experiences/esfehani.mohamad3@gmail.com/gear-vr/849609821813454
- Splitt Martin. 2016. Martin Splitt - A journey to WebVR. NLHTML5 - front-end meetup group, Amsterdam, Netherlands. (26 Jule 2016). <https://www.youtube.com/watch?v=8IXFCIbOgmQ>
- Qian Feng, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan. 2016. Optimizing 360 Video Delivery over Cellular Networks. 1–6. DOI: 10.1145/2980055.2980056
- WebVR Rocks. 2017. WebVR Rocks: Your guide to VR on the Web. (2017). <https://webvr.rocks>
- Zheng ,Vincent. 2017. Web VR is The Future Virtual Reality Content For Browser. (2017). Retrieved from <https://blog.visualpathy.com/what-is-webvr>