

TECHNOLOGY REVIEW AND IMPLEMENTATION

CS461 - FALL 2016

Charles Siebert, Branden Berlin, Yipeng "Roger" Song

February 17, 2017

Abstract

The technology of Virtual Reality (VR) currently is not cost effective to today's market, as the cost of high-end setups required makes it difficult to afford. Browser developers are focusing primarily on expensive high-end high-performance hardware over mobile devices for Augmented Reality (AR) or Virtual Reality (VR) on the web. Doing AR/VR on the mobile web allows more developers to enter the field and deliver to more customers. To accomplish this, we are working on a project called Mobile AR/VR Performance, which focuses on researching to profile and identify performance bottlenecks in 3D web content on mobile devices. We will file issues in the open source projects for Chrome, Firefox through A-Frame and Three.js to determine and identify those bottlenecks. We hope to accomplish this by reporting the challenges and opportunities for performance VR/AR applications, and write a blog post detailing the project results and their best-practices.

Contents

1. Introduction	1
2. Generation Of Scenes In Browser	1
2.1 Options to Use	1
2.2 Evaluation of Options	1
2.3 Options Comparison	1
2.4 Discussion	1
2.5 Selection	1
3. Processing/Analyzing Data	1
3.1 Options to Use	1
3.2 Evaluation of Options	1
3.3 Options Comparison	2
3.4 Discussion	2
3.5 Selection	2
4. Setup Local Host Server	2
4.1 Options to Use	2
4.2 Evaluation of Options	2
4.3 Options Comparison	2
4.4 Discussion	2
4.5 Selection	2
5. Generation and capture of the data	2
5.1 Options to Use	3
5.2 Evaluation of Options	3
5.3 Discussion	3
6. Generation of Scenes on a Mobile Device	3
6.1 Options to use	3
6.2 Evaluation of Options	4
6.3 Discussion	4
7. Version Control and Documentation	4
7.1 Options to Use	4
7.2 Evaluation of Options	4
7.3 Discussion	5
8. Creation of Scenes in A-frame (Roger)	5
8.1 Options to Use	5
8.2 Evaluation of Options	5
8.3 Options Comparison	5
8.4 Selection	5
9. Development Tools (Roger)	5
9.1 Options to Use	5
9.2 Evaluation of Options	5
9.3 Options Comparison	6
9.4 Selection	6
10. Language Tools (Roger)	6
10.1 Options to Use	6

10.2 Evaluation of Options	6
10.3 Options Comparison	6
10.4 Discussion	7
10.5 Selection	7
References	7

1. Introduction

Our project "Optimizing Virtual Reality and Augmented Reality Performance on Mobile Web Applications," nicknamed "OVRAR" is the development of a VR application and analyzing the performance based on the restrictions of using it in a mobile environment with the same devices. Using the data we collect from the application, we can determine specific bottlenecks and performance issues that may be incurred on the device based on different test implementations. This document outlines the overview of technologies that will be used during the lifeline of the project. Each technology listed will have three tools that could be used to accomplish the specific task, and a comparison between them that will lead to a conclusion of which tool was chosen, and why. The purpose of this document is to break down each section of the project that is required to determine how we will implement the program and how the data will be analyzed.

Every three sections will be authored by each of us, leaving us 9 sections of technologies to be listed. The first three (Section 2 through Section 4) is written by Branden, the second three (Section 5 through Section 7) is written by Charles, and the last three (Section 8 through Section 10) is written by Roger. Each person authoring their sections will ultimately be responsible for those technologies they discuss and conclude to use to accomplish the tasks of the system and the capturing of data.

2. Generation Of Scenes In Browser

2.1 Options to Use

The options for browser scene generation are: Internet Explorer, Google Chrome, Mozilla Firefox, Opera, and Safari.

2.2 Evaluation of Options

Each of the web browsers are capable software applications for retrieving, presenting, and traversing information resources that are not only identified by a URL but an IP address as well and are capable of hosting a web page, image, video, or other piece of content such as 3D imaging.

2.3 Options Comparison

While each of these options are testable web browsers, they may require different coding aspects and each option might require different libraries or coding tactics (i.e. Explorer does not like certain HTML and CSS variable phrases and structures that chrome and firefox do) and would require extra time and effort to transition code. That being said, it would allow for more robust research should we be able to verify the bugs we encounter through multiple browsers.

2.4 Discussion

Though we may be able to test in each of these browsers, our client has asked us to use Mozilla Firefox because it is the browser environment that our research project is based around and that is the company our client works for. However, once we encounter bugs, we will attempt to verify them in multiple browsers to get a better understanding of why that error arose in Firefox and if it is singled out there, or expands across multiple browsers.

2.5 Selection

Mozilla Firefox

3. Processing/Analyzing Data

3.1 Options to Use

Options for the recording and analyzing of data are: Excel, Google Sheets, Apache OpenOffice, LibreOffice, and many more.

3.2 Evaluation of Options

Each of these tools are used for analyzing and recording data and some over others are better at manipulating data to either find patterns of present trends. Excel and google sheets are simple computer software programs that are used for storing, organizing and manipulating data. Apache OpenOffice is a software suite for word processing, spreadsheets, presentations, graphics, databases and more.

3.3 Options Comparison

Though each of these options are capable of recording general data, some of them offer more features than others and some are more complicated than this project's data recording needs to be. Apache OpenOffice and Libre office, for the time being, seem beyond the scope of our data recording needs. The type of data we are going to be recording analytics and number comparisons, which Excel and Google Sheets are perfect for.

3.4 Discussion

So without needing fancy data analytics tools, we simply need a place to store the data we record which will be software and hardware specs. We are going to need to store large quantities and have them readily comparable against different parameters, which the spreadsheet aspect of both Excel and Google Sheets provide. That being said, Excel offers more design, formatting, and general mathematical capabilities should we require them in the future.

3.5 Selection

Excel

4. Setup Local Host Server

4.1 Options to Use

The options for setting up localhost servers are: Nginx, Apache, and Lighttpd, or OSU's stak server.

4.2 Evaluation of Options

The first three are most used free web-servers provided by the open source community, making up over 50% of domains, that will allow of to remotely connect via mobile web-browser through IP address. The stak server is a server already implemented for engineering students to host their own personal webpages.

4.3 Options Comparison

a) Nginx, the second most popular web hosting server software supports most OS and uses event driven architecture instead of threads. b) Lighttpd is the third most used server software but serves less than 1% of domains and is also event-driven. c) Apache is the most used server software with it being used on 60% of domains and it supports a multitude of packages. Additionally it is easier to install and provides for a much richer and reliable environment. d) Stak Server is the most readily available and allows remote connect. No setup needed.

4.4 Discussion

Because we don't know the situations we are going to run into through our web hosting, we are going to want as server host that is not only more reliable and easier to use, but we are potentially going to want to apply different packages, and because apache holds the majority of user use, we can trust that it will be able to cater to our needs and have more up-to-date features. Additionally, the extensive documentation already online regarding Apache will potentially help us in troubleshooting the errors we run into and will allow for much quicker and hassle-free fixes. (*UPDATE*)We have since discovered that OSU has a server platform already implemented for each student and it is remotely connectable for on-the-fly changes. Because of this and in addition to Apache-to-mobile issues, we have switched.

4.5 Selection

Stak Servers

5. Generation and capture of the data

The biggest part of our project is the generating the data we receive in our test cases, and capturing it so we have good and reliable metrics of performance data to analyze. The challenge that arises from this is when dealing with multiple frameworks of development, such as running this project on the Android OS, on a phone, where it renders multiple test cases through two browsers that are interpreting languages through two different frameworks. If performance issues occur during development, we need to be able to understand where the bottleneck occurs, whether it lives in the OS, the browsers, or the frameworks we're developing on.

5.1 Options to Use

There's a couple of options to use, specifically we're looking at the Android SDK, which has features that are included with the capturing of data through different levels of the android stack, where there's performance metrics with the phone, the operating system, and within the browser. A competing tool is the Firefox Dev tools, which allows the connection of devices over a network, and will capture performance metrics through the device onto the browser. This holds information such as performance usage, memory consumption and, and rendering speeds of the scene. Chrome Dev tools has similar, competing tools to analyze the metrics when being used to render the scenes on Chrome.

5.2 Evaluation of Options

Below is a list of functionality that comes with each of the tools (Android SDK [2], Firefox Dev Tools [3], Chrome Dev Tools [4]) to be able to compare and contrast the tools that we will use when generating our test scenes through this project. Each of these tools have more features than what is listed, this is just a short description of what each tool will be useful for us.

Android SDK	Firefox Dev Tools	Chrome Dev Tools
Performance Monitor: Allows us to track the memory and CPU usage during rendering, and frames being developed. Memory Dump: Allows us to see memory allocations to the device to see where memory leaks occur. GPU Profiling: An option that allows us to track the time it took to draw the last 128 frames.	Remote Debugging: Firefox allows us to do remote debugging of code that is running in Firefox for Android over a USB connection. Performance Tools: Has multiple tools to analyze processor and memory usage, frame rates, and call trees to see how performance is handled in the browser compared to Android OS. Source Editor: Allows us to make quick edits to our source code during runtime and see performance or visual changes.	Remote Debugging: Similarly to Firefox, allows remote debugging of the code that is running in Chrome for Android over a USB connection. Timeline Performance: Chrome has a tool that will record and analyze all the activity in the browser as it runs, capturing FPS and CPU usage, along with profiling the JavaScript stack. JavaScript Debugging: Explicit tools for JavaScript debugging, allowing us to step through the code, sets breakpoints at predetermined intervals, and watch variables during runtime.

5.3 Discussion

These tools will all actually be used simultaneously, as they are recommended by our client to test multiple tools as we conduct our research. The tools all have a set of performance analysis built into them, which allows us to be able to capture the performance metrics while we test in multiple environments. Each tool has special features specific to each browser, which will help during debugging, and pinpointing areas that are causing performance issues that may occur in different areas of rendering (OS level, browser level, or JS/interpreter level). More tools allow us to have a wider set of test cases to determine possible areas of optimization issues and bottlenecks.

6. Generation of Scenes on a Mobile Device

How the scenes will be rendered, and the choice of which device was predetermined for us already. Our client informed us that it would be beneficial to testing if they were all conducted on the same device, the same device used with other teams that are similarly working on the A-Frame framework. This reduces the amount of variables in testing, especially when the devices has different specifications and may handle scenes better or worse than the devices they use. It helps us pinpoint specific areas in testing where bottlenecks occur when attempting to render the scenes on the phone.

6.1 Options to use

Since our client requires us to use a specific mobile device to have the same device and have similar testing case for our research, the Nexus 5X is one of the newer phones that came out last year, and has substantial hardware specifications to handle the rendering of graphical scenes. As the project is developed on the A-Frame framework, and implemented through the Firefox and Chrome browsers, the Android OS platform is the most supported for this kind of testing and development. Safari and iOS are excluded from this testing due to issues with A-Frame running on Safari, that research is outside of the scope of our project, and because of this, iPhones were obviously excluded as a choice of hardware for development purposes.

6.2 Evaluation of Options

The only option is the Nexus 5X, as this is what our client requires us to use for testing development. Though for the sake of comparison [5], this device is best for our testing development because it allows a wider range of test environments due to the powerful hardware specifications and the platform it was built on. In the comparison below, the hardware specifications the newest generation iPhone 7 and the LG Volt device will be compared.

Nexus 5X	iPhone 7	LG Volt
OS: Android 7.0 "Nougat" Battery: 2,700 mAh Lithium Battery RAM: 2 GB DDR3 Memory CPU: 1.8 GHz Hexa-core 64-bit Processor GPU: Adreno 418 @ 600 MHz	OS: iOS 10.1.1 Battery: 1,960 mAh Li-Po Battery RAM: 2GB DDR4 Memory CPU: 2.34 GHz Quad-core 64-bit Processor GPU: Apple G9	OS: Android 5.0 "KitKat" Battery: 3,000 mAh Li-Ion Battery RAM: 1 GB DDR3 Memory CPU: 1.2 GHz Quad-core 64-bit Processor GPU: Adreno 305 @ 450 MHz

6.3 Discussion

As shown in the table, the Nexus 5X is a very capable phone to handle the rendering of graphical scenes through the browsers. Compared to the other choices shown, it is stronger or similarly comparable in hardware specifications to the newer iPhone, and the LG Volt. Along with uniform testing platforms, the hardware on the Nexus 5X will be reliable enough to generate substantial test cases that can be used on future iterations of mobile devices as the hardware becomes stronger and more efficient. It's logical to test on best hardware for the time, as the hardware will become deprecated within a couple of years.

7. Version Control and Documentation

Our deliverable product for this project will be a report detailing the results we've found through our sets of test cases, and so having proper version control and documentation is a key thing to have for this project. We have a need to be able to have information readily accessible to us for when we are dealing with multiple test cases, to be able to review previous results and data to be able to determine the cause of possible bottlenecks that occur in real time. We need to be able to reference our findings, import and export the data in a fashion that would not mess with the formatting of our documentation and handle imported into our choice of version control without issues.

7.1 Options to Use

In terms of version control, we are going with Github, as it is a requirement for the class. Everything that is developed, recorded, written, and analyzed will be uploaded to Github. This will encapsulate the entirety of our project. Our documentation can be accomplished a couple of ways, either through LaTeX, Microsoft Word, or Google Docs. Since our documentation is being uploaded to Github, it's essential that our choice of tool meets the requirements of being easily accessible, not ruin formatting, and imported/exported easily into our version control. Each of these tools have their own purpose, and will be detailed below.

7.2 Evaluation of Options

Microsoft Word is probably the most widely used version of word processing, but has issues when importing and exporting to Github. Since Github essentially parses line per line and uploads the differentials of added/deleted lines every time the document is committed, this makes it difficult to understand what lines were changed in the document. Also when attempting to importing data tables and graphics into Word, it will make formatting as an unnecessary chore.

Google Docs is a free version of version control of documentation and word processing all in one, but comes with a caveat of only really working within itself. Attempting to export the files processed in Google Docs will mess up with formatting, and will not be easily parsed into our version control, where needing to see up to date documentation will become even more unnecessary work. While it does attempt to create a form of version control, it only adds to being an extra place to "check" for up to date information, and the word processing formatting for it just does not fit within our scope for this project.

LaTeX is a mark-up language, similar to HTML. As a word processor, it breaks down the formatting based on the packages included into the file, and the defined parameters allow uniform formatting in all specifically defined areas. Even graphics will be easily imported into the document, as tools such as PSTricks allow the rendering of the images based on a few function calls in the document, and when compiled it will generate the file properly. This fits in the scope for our project as it will fit within the use of our Github version control, and will allow us to see the changes made to the document and view them in real time. It's also very trivial to add special tables of data or analytical images as it is just a file with marked up text where formatting is already predefined.

7.3 Discussion

Documentation is done very trivially and the compiler for LaTeX handles formatting for everything else. The results are very predictable and the ability to view the contents of the actual .tex file and the resulting .pdf file allows for easy portability and readability into the version control we are using. LaTeX makes it also very trivial to import tables and images into it, when it comes time to generate the analyzed data that we receive for this project. Every other option we discussed has some unnecessary challenges when implementing our results into our documentation. They have too many downsides, while LaTeX fits into the version control scheme very easily.

8. Creation of Scenes in A-frame (Roger)

8.1 Options to Use

As the purpose of this project is to determine areas of development within A-Frame where practices will be best used, so we are going to use A-Frame to create scenes. A-Frame is an open-source WebVR framework for creating virtual reality (VR) experiences with HTML with the use of the Three.js framework.

8.2 Evaluation of Options

A-Frame will handle the specific Three.js calls to help render the scene with 3D objects with the use of WebGL API. The WebGL API utilized by Three.js will interact with the hardware processing when drawing the topology of polygons, textures, lightings, animations, and scenes to the viewport. There are three things about A-Frame that are important to our project:

- a) A-Frame is a framework for reworking Three.js implementations to make the calls to draw 3D scenes simplified by using HTML.
- b) A-Frame Takes the marking A-Frame tags and produces the proper JavaScript calls to generate scenes to the browser.
- c) These calls are the input information to the Three.js framework, which outputs the polygons and scenes to be drawn.

8.3 Options Comparison

A-Frame is the only option we have, since it is the requirement of the project, and not other frameworks fit in our scope of the project.

A-Frame is an open-source WebVR framework for creating virtual reality (VR) experiences with HTML. We are able to build VR scenes that work across smartphones, desktop, the Oculus Rift, and the room-scale HTC Vive.

8.4 Selection

Based on all the points mentioned above, we are going to use A-Frame to create scenes.

9. Development Tools (Roger)

9.1 Options to Use

In order to develop a software project as a group, one biggest point is to choose the right development tool. There are a few good development tools for software, such as Visual Studio, Notepad++, Sublime, and so on. We will evaluate and compare with one from another, and decide which is the best choice for this project.

9.2 Evaluation of Options

Visual Studio offers a comprehensive development environment to help Web developers build standards-based Web applications and services. It improves productivity by allowing users to rapidly develop, test and deploy Web solutions. Additionally, Visual Studio offers a free version aimed at Web developers. With Visual Web Developer Express, you get a full featured web development environment for working with ASP.NET, Javascript and Web standards.

One of the main advantages of Notepad++ is that it requires no training and you can start HTML straight away. It comes free with every version of Windows which means you do not have to go through the hassle of downloading the program. The program doesn't take up much space on the laptop/desktop PC and because its low specification software it means that it doesn't take a long time to load.

Sublime has got a modern and colorful interface. It also has full screen support, and uses the Minimap feature pretty well which makes you deal efficiently with files carrying thousands of code lines. Plus, distraction free Mode that makes you more focused on Another advantage of sublime is it can auto saving of your code to keep you worry less and the Spell Checker doesn't let you make errors while writing codes. Besides, availability of Multiple Selections will eventually simplify your task. One big disadvantage of sublime is that its license costs \$70, which makes it an expensive deal than many other code editors

9.3 Options Comparison

The goal of Visual Studio is to provide rich development tools to all developers globally on any platform. Development teams will be able to develop software for Web, mobile, server and desktop with Visual Studio. Also, it also has an online version, Visual Studio Online, which is a set of tools that makes it much easier for continuous integration across different platforms.

Of the free options here, Notepad++ is extremely popular. It is often mentioned in blog posts as a great free option for someone just getting into code editing or who needs to edit a few files here and there (but who probably don't want to buy an expensive or complex editor). Notepad++ is a wonderful, simple option not just for beginners but developers at any level.

Sublime text is a beautiful, feature rich option for code writing and editing. A big draw for this editor is the fact that it puts a premium on user experience. This includes features like distraction free writing mode, quick shortcuts/search, split editing, and so on.

9.4 Selection

In this project, we are going to use Notepad++ as the development tool as it allows us to make quick edits to our files so we can push them to our stack servers without having to run any compilations.

10. Language Tools (Roger)

10.1 Options to Use

An implementation language is a formal computer language or constructed language designed to communicate instructions to a machine, particularly a computer. Understanding difference(s) between programming languages is crucial. If wrong language is chosen for a project, it will take a lot of time and efforts to change the course and re-implement the project or its part in different language. For our project, since we are going to test the performance of web app through A-frame, which is a Web framework, so we should choose from the languages that are specific to web development. A few examples are HTML, Java Scripts, and PHP.

10.2 Evaluation of Options

HTML is easy enough to write, but errors can be costly. Another disadvantage is the time it takes to choose the color scheme of a page and to create lists, tables and forms. An advantage to HTML is that it is easy to code. Much of the code can be customized by someone who knows proper HTML formatting. HTML also allows the use of templates, which makes designing a web page easily.

JavaScript is very fast because any code functions can be run immediately instead of having to contact the server and wait for an answer. Plus, JavaScript is relatively simple to learn and implement, and it plays nicely with other languages and can be used in a huge variety of applications. Additionally, being client-side reduces the demand on the website server. However, since JavaScript code executes on the users' computer, in some cases it can be exploited for malicious purposes. This is one reason some people choose to disable JavaScript. JavaScript is sometimes interpreted differently by different browsers. Whereas server-side scripts will always produce the same output, client-side scripts can be a little unpredictable.

One advantage of PHP is that it has both procedure programming language and OOP language features. This mainly attributes to its evolution along the way. This means that programmers from different programming language background can pick up this language within short period of time. Its disadvantage also comes from its ample language features. Some libraries written by a programmer from a procedure programming language may be difficult for programmers with an OOP background to maintain.

10.3 Options Comparison

HTML, stands for Hyper Text Markup Language, is the standard markup language for creating Web pages. It describes the structure of Web pages using markup, and its elements are the building blocks of HTML pages which are represented by tags. HTML tags label pieces of content such as "heading", "paragraph", "table", and so on. Browsers do not display the HTML tags, but use them to render the content of the page

Programs written in JavaScript run in the web browser itself, so if the website has a JavaScript program, the program will be automatically fetched by your visitor's browser and executed on his/her computer.

PHP programs, on the other hand, run on the computer where your website is located, that is, on your web host's computer. After the PHP program does what it needs to do, it sends the result to the visitor's web browser, which merely displays the results.

10.4 Discussion

As this project serves as to be used as research, correctness will be based off of how well we can push the boundaries of our software, based on the performance the hardware can provide. Therefore, we need to collect the information in different types of implementations between scenes the program generates, so we should be able to understand and write some test scripts.

10.5 Selection

We are going to use HTML, JavaScript, and PHP simultaneously, as all three of them are very crucial to understand our project. We need to understand these languages in order to find the bottlenecks, where the optimization needs to be done.

References

- [1] Authors of Ref1 *Ref1 Title*. Ref1 Publisher
- [2] Vogella. *Android Application Performance Profiling through Android SDK*.
<http://www.vogella.com/tutorials/AndroidTools/article.html>
- [3] Mozilla Developer Network. *Firefox Developer Tools*.
<https://developer.mozilla.org/en-US/docs/Tools>
- [4] Google Chrome DevTools. *Chrome DevTools*.
<https://developers.google.com/web/tools/chrome-devtools/>
- [5] GSM Arena. *Phone Specifications*
<http://www.gsmarena.com/>