OPTIMIZING VIRTUAL REALITY AND AUGMENTED REALITY ON MOBILE WEB APPLICATIONS - REQUIREMENTS DOCUMENT

CS461 - FALL 2016

Charles Siebert, Branden Berlin, Yipeng "Roger" Song

November 4, 2016

Abstract

The technology of Virtual Reality (VR) currently is not cost effective to the today's market, as the cost of high-end setups required makes it difficult to afford. Browser developers are focusing primarily on high-end specialized hardware for VR on mobile, and are generally ignoring Augmented Reality (AR). Therefore, doing AR on the web allows far more developers to enter the field. To accomplish this, we are working on a project called "Mobile Web App", 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

This project requirements document outlines the entirety of the project, including the necessary definitions and references used throughout the project's duration as well as within the documents created in the project. This paper focuses on what the project is, it's purpose, and the problem it is trying to solve. The problem we need to solve is to optimize performance issues found within the A-Frame web framework, and determine how well it works on mobile devices. The software we develop for this project will be used as a series of test cases to measure the performance differences between different types of implementations, or specific scenarios in implementation where we mean to pin point bottlenecks that would occur within mobile devices. After collecting the necessary data (frame rate, processing consumption, battery life-span, and hardware limitations), we will compile a report that analyzes the information to determine the best practices, or bugs that may occur during development. The software we are developing will be conducted on a Nexus 5X phone, which is the standard of phone being used by other developers working on A-Frame. This helps to eliminate the possibility of hardware differences when measuring the metrics of performance data and bug-specific problems relating to hardware.

1.1 Purpose

The purpose of this project is to determine areas of development within A-Frame where practices will be best used, as they will least be likely to impede on bottle necking either the software or hardware whe n optimizing the software for performance. This project is focused towards the advancement of an open-source, developing web framework, and the developers making their own products with A-Frame and for mobile devices. The developers will be using our project research as a means to avoid these bottlenecks in this evolving environment.

1.2 Scope

Optimizing VR and AR for Mobile Web Apps is to determine Virtual Reality (VR) and Augmented Reality (AR) bottlenecks that exist in mobile devices within the A-Frame framework. The bottlenecks can be caused from either unoptimized development of software, underpowered or unoptimized hardware found in existing devices, or potential bugs or limitations found within the framework itself. The software itself, which is developed on A-Frame, will generate multiple scenes where it will test the graphical capabilities of the hardware within the mobile devices, the types of different implementations of certain scenes, and determine areas of optimization through these multiple scenes. The software will be used to create a report that will analyze the information collected about processing power, frame rates, battery usage, and the limitations of the framework to determine the best practices for implementing more graphically intensive programs on A-Frame.

Developers other than us will use the information in the report to determine the best way to approach at designing their programs, as the software we create will only serve as test cases and stress testing for mobile devices to collect this information.

1.3 Glossary

1) **Virtual Reality (VR):** Computer generated three-dimensional environment that immerses the user into the environment using special equipment or implementation techniques.
2) **Augmented Reality (AR):** Provides a composite view to the user based on computer generated environments super-imposed onto the view of the real world.
3) **Operating System (OS):** Software that supports an interface to support computer's basic functions, such as process scheduling, executing tasks, and allowing user interface. Specifically this project is in regards to Android OS.
4) **Web Framework:** A software framework that is designed to support the development of web applications including web services, web resources and web APIs.
5) **A-Frame:** A-Frame is an open-source WebVR framework for creating virtual reality (VR) experiences with HTML with the use of the Three.js framework.
6) **Three.js:** JavaScript framework that allows accessible development WebGL applications.
7) **WebGL:** JavaScript API that allows for rendering interactive 3D and 2D computer graphics within any compatible web browser without the use of plug-ins.

8) **Viewport:** A viewport is a viewing region in computer graphics. This region is defined in the software to allow for viewing the drawing of objects. This viewport is typically bounded by a window or by the full screen of the mobile device.

9) **Local Web Server:** A webs erver for hosting web page content to allow access on local networks, without having to go out into the internet to access the information.

10) **Hosted Web Server:** A server that is hosted externally on the internet, where it holds and displays the web information, needing to go out to the internet, and back again to receive the proper information.

11) **Implementation Languages:** a formal computer language or constructed language designed to communicate instructions to a machine, particularly a computer. i.e. HTML, JavaScript, C, C++, etc.

12) **Mobile Devices:** A device that is able to be held and portable by a user, typically a smart phone or table.t

13) **Rendering:** Part of the graphical process that draws everything into the "view's" scene. This includes textures, animations, objects, surface information, etc.

14) **Bottleneck:** An effect impeding on the rendering process, which occurs between the hardware and software aspects.

15) **Optimize:** Process of making something as fully functional or effective as possible, including the types of limitations that occur in the environment.

16) **Performance:** The process of how well the software handles the action or function of the software.

17) **Software Bug:** An error, flaw, failure or fault in the system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

1.4 References

No references has been used in this document, yet.

1.5 Overview

This paper is a requirements document that describes the approaches we take during this project. It is separated into three different clauses. The first (next section) clause talks about the project description, and introductory information on the constraints and interfaces within the project. The second clause discusses the exact requirements of the project, and their following subsections in detail. The final clause, at the end of this document, will have a Gantt Chart provided, which outlines the time frame at which we will progress through this project. It details the type of work we are doing in roughly one to two week sprints, and when they occur.

2. Project Description

This project is to determine areas of optimization within the frameworks specified, by which a software program will be developed to test specific graphical scenarios to stress test the kinds of interfaces and constraints listed. The report will be used to improve the development process of VR and AR application on Mobile Web applications, as they are heavily underdeveloped compared to desktops due to their lack of processing power and high battery consumption.

2.1 Project Perspective

This software is developed in a the A-Frame framework, which consists of development using HTML and JavaScript implementation languages to manage the graphical scenes. A-Frame utilizes Three.js framework, which allows A-Frame to develop WebGL content within a browser, and reduces boilerplate programs that are typically found in graphic programs. Our project will use this platform to develop a culmination of stress testing scenes on specific hardware devices (Nexus 5X) that are the current standard for developing internally.

2.1.1 Operations and User Interfaces

The extent of user interaction of this software will be determining performance characteristics and moving around the scenes, either by touch or walking. The interface in which the user will interact with the program will be through a 5 inch (diagonal) size screen mobile device. The software will be accessed through a web browser that is hosted on either a local address (only for testing purposes) or hosted on a specific website.

2.1.2 Hardware Interfaces

The software will be used and tested on mobile devices that has a 5.2 inch screen at 1920x1080 resolution. This mobile device uses a Hexa-core processor (Snapdragon 808) clocked at 1.4 gigahertz (GHz), an Adreno 418 graphics processor, 2 gigabytes (GB) of RAM, and a 2700 mAh battery. This hardware is identified and quantified to determine the areas of which bottlenecks occur, and measure the metrics related to the hardware and their uses.

2.1.3 Software Interfaces

The software will be developed and tested on an Android Operating System, version 7.0, nicknamed Nougat. The phone will then utilize Mozilla's Firefox and Google Chrome web browsers to use as the testing interface. The browser allows the development of graphical scenes within A-Frame with WebGL and Three.js to allow us to test the framework and stressing the software and hardware. The browsers mentioned will be constantly up to date with the current version on the Google Play Store.

2.1.4 Communication Interfaces

There is no communication interface that will be developed within this program. This will act as a standalone program to serve the function of determining areas of optimization and bugs.

2.1.5 Memory Constraints

Potential constraints from development on the Nexus 5X will will be encountered from either the types of implementation in the program, the quality, or amounts of texturing and drawing object to the scene; it is dependant on how complex the implementation becomes. The mobile device will only have 2 GB of on-board RAM, meaning a more restrictive work load to test with. Additionally, drawing higher-quality textures with regard to the amount of textures, along with complexity of the scenes drawn, will determine whether or not there will be memory constraints and will be one of the metrics measured to determine this as an area of bottleneck.

2.2 Project Functions

The key points of this project are candid, as this project primarily pertains to research and mid-development discoveries. Though we are developing a program, it isn't the main purpose of this project; the program is supposed to supplement our research towards our final report at the end of this project (outlined in Gantt Chart). The projects functions are organized in a list shown below:

1) The software is supposed to be runnable in a mobile environment.
2) It will determine bottlenecks between the software and the hardware.
3) It will be used to measure frame rates, processing, RAM, and battery consumption.
4) Development will result in us reporting bugs through A-Frame with Firefox to Mozilla's bug reporting platform.
5) It will help us determine the best practices for some graphical implementations of the A-Frame framework in a VR environment.

2.3 Project User Characteristics

The intended users for this research project are advanced software developers. The developers should have a general understanding of the work flow with development on mobile devices, and an advanced understanding of the program's frame rates and battery usage. These developers should have an intermediate background experience with developing on mobile devices, experience in generating and rendering graphical screens on devices through the view port, and a general understanding of what bottlenecks are and how to identify them in order to use this research. Without being able to identify what kinds of bottlenecks exist in their own program, they won't be able to use our research to properly optimize their issues.

2.4 Project Constraints

The biggest constraint is implementation on mobile devices, as these end up typically being weaker than normal hardware that is used for similar, tough, unoptimized graphics rendering that desktop computers are able to provide. The specific hardware specifications are listed and shown as a constraint in section 2.1.2. Aside from the hardware constraints shown, there's a high-order language requirements to be used (A-Frame and it's HTML markup) as this

developing framework still has optimizations patches to undergo. The optimization issues or bugs that we do find during our project will be constrained by what A-Frame will allow us to do with software in terms of graphical processing and rendering of scenes. The higher-order control functions from the Three.js and A-Frame framework acts as this constraint, as it will limit what we can develop in the scenes and how we approach the design of the scenes.

2.5   Project Dependencies

This project is to be developed regarding the software constraints detailed in section 2.1.3. Updated versions of Android (versions 7.0) should not affect the development of this project, or any future iterations of it. This project is also dependant on the frameworks, Three.js, which handles drawing the scenes with WebGL, and A-Frame, which reduces the amount of boiler plate for graphical development, allowing implementation of drawing the scenes through HTML markups. Future updates to these frameworks could potentially alleviate optimization issues that we outline, or bugs that we report.

This project is to be tested against the newest versions of Firefox and Chrome, which will become a dependency due to us not testing development on other platforms such as the OS iOS, or the browser Safari. As we are run into issues during development, we will report the bugs to the respective camps if they become related to a browser-specific issue. JavaScript is an interpreted language while HTML is just a markup language, which are easily understood by these two browsers. Any updates to these browsers will potentially alleviate issues reported.

3.   Specific Requirements

This section outlines the specific requirements of the project, which will include information on the external interfaces, functions, performance requirements, and the underlying attributes of the project. The problem this project has is that we don't know exactly what problems we will run into, how many problems there are, or how we will work around them. The solution of this project lays in the problems we do find, such as the software bugs or optimization issues.

3.1   External Interface

There are three parts of the software that will have different kinds of external interfaces, and how they interact with the different types of software. The first one is the browser, which will interact with the A-Frame development project that is hosted on the web server. 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.

1) Browser (Firefox or Chrome)
   a) Acts as the interpreter of the languages of HTML and JavaScript when accessing the web page where the project lives.
   b) Takes the HTML markup and JavaScript functions on the web server and displays images in the viewport.
   c) Will take up the entire screen size (5.2 inches) when viewing and testing on a mobile device.
   d) Commands will be input through the screen or movement physics of the phone itself. This interacts with the scene and changing what is displayed on the viewport.
2) A-Frame
   a) Framework for reworking Three.js implementations to make the calls to draw 3D scenes simplified by using HTML.
   b) 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 scenese to be drawn.
   d) The commands to draw the scene reduces the boilerplate implementation of WebGL, with single lines like `<a-scene>`.
3) Three.js
   a) JavaScript framework to provide interaction with the WebGL API and is attached with A-Frame commands.
   b) The input comes from the A-Frame calls, which act as the parameters in HTML, to the Three.js function calls which will output the scene changes to the viewport.

c) The outputs are related to the hardware at this stage, as they interact with the hardware, where the polygon vertices, normals, texture, and their coordinates exist in memory on the device.

d) The output will return to the browser a scene that is drawn to the viewport.

e) Messages will appear if errors occur, but will otherwise output the software to the broswer.

3.2   Functions

The software will have specific actions that will interact between the user and the system. This will involve all the specific actions that will occur within the system if and when a user chooses an action. These actions include

1) The system shall interact with a local web server or a hosted web server.
    a) The system shall allow the user to see the content on the web server.
    b) The system shall allow the user to interact with the scene by moving around or touching the screen.
2) The system shall handle the rendering of multiple scenes.
    a) The system shall render the scenes in with differing implementations analyze those performance metrics.
    b) The system shall attempt to draw partial scenes if a software bug occurs.
    c) The system shall handle rendering errors with proper error checking messaging to the screen if a scene cannot be rendered.
3) The system shall be used to read in performance metrics as the software progresses through different scenes and implementations.
    a) The system shall be used to measure the performance of frame rates, memory consumption, processor consumption, battery consumption, and response times on the device being used.
    b) The system shall be an example of our concepts of best practices in developing in the A-Frame framework.

3.3   Performance Requirements

While there are minimum requirements to attain a responsive system, this project strives to obtain optimized performance through the framework. There are multiple types of performance metrics and requirements for us to attain a fast and responsive system while optimizing the performance of the system. The type of hardware that we are using, detailed in section 2.1.2, will be the hardware specifications that we are examining while stressing the software program. We are defining "a fast and responsive" system to be at a minimum:

1) The software will be ran and tested on the mobile device used for this project, the Nexus 5X.
2) Response time from user interaction to software processing information to be .1 second (about 3 frames of input delay).
3) The software will display 30 Frames Per Second (FPS) to the phone through the viewport.
4) The software will not exceed the amount of on-board RAM (2GB) during the time of drawing and texturing the scenes.
5) The processors will not exceed the maximum usage (100%) while the device is rendering the scene.
    a) Due to possible hardware bottlenecks, if the processor usage does exceed 100%, it will still allow the viewing of the scene with the aforementioned requirements.
6) The software will not exceed excessive battery usage. The Nexus 5X has a total capacity of 2700 mAh.
    a) Excessive battery usage will be measured how much battery capacity is used per hour.
    b) The battery should at the least last three hours on a full charge while undergoing the tests of this program.
7) The software will not produce excessive heat through hardware being stressed by the software.
    a) Excessive heat is defined to be a high enough temperature to trigger the phone's internal underclocking feature to reduce the temperature.
    b) Documentation, at this time, cannot be found at what temperature this occurs, but the temperature of the phone and the components will be tracked and documented if it causes the aforementioned requirements to not be met.

3.4   Database Requirements

This software will not make use of a database, as there is no functional requirement and does not fit our scope.

3.5 Attributes And Correctness

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. By collecting the information in different types of implementations between scenes the program generates, it will allow us to analyze the data to compare and contrast the information based on our performance metrics detailed in Section 3.3. The better the performance metrics, the better, as that is the purpose to optimize the performance of the software that utilizes the A-Frame framework. Our correctness will be based off of the amount of software bugs that we encounter, and the collection of this data that will ultimately determine what the best practices for implementation are, and bear as proof.

3.5.1 Availability

Availability is large due to the immense market that the mobile devices provide. The software has to be responsive to human interactions, which would include tools to respond to user input, and changing the view of the scene based on the view-port and position the phone provides.

3.5.2 Security

Not applicable, as this is a proof of concept, this does not lie within our scope.
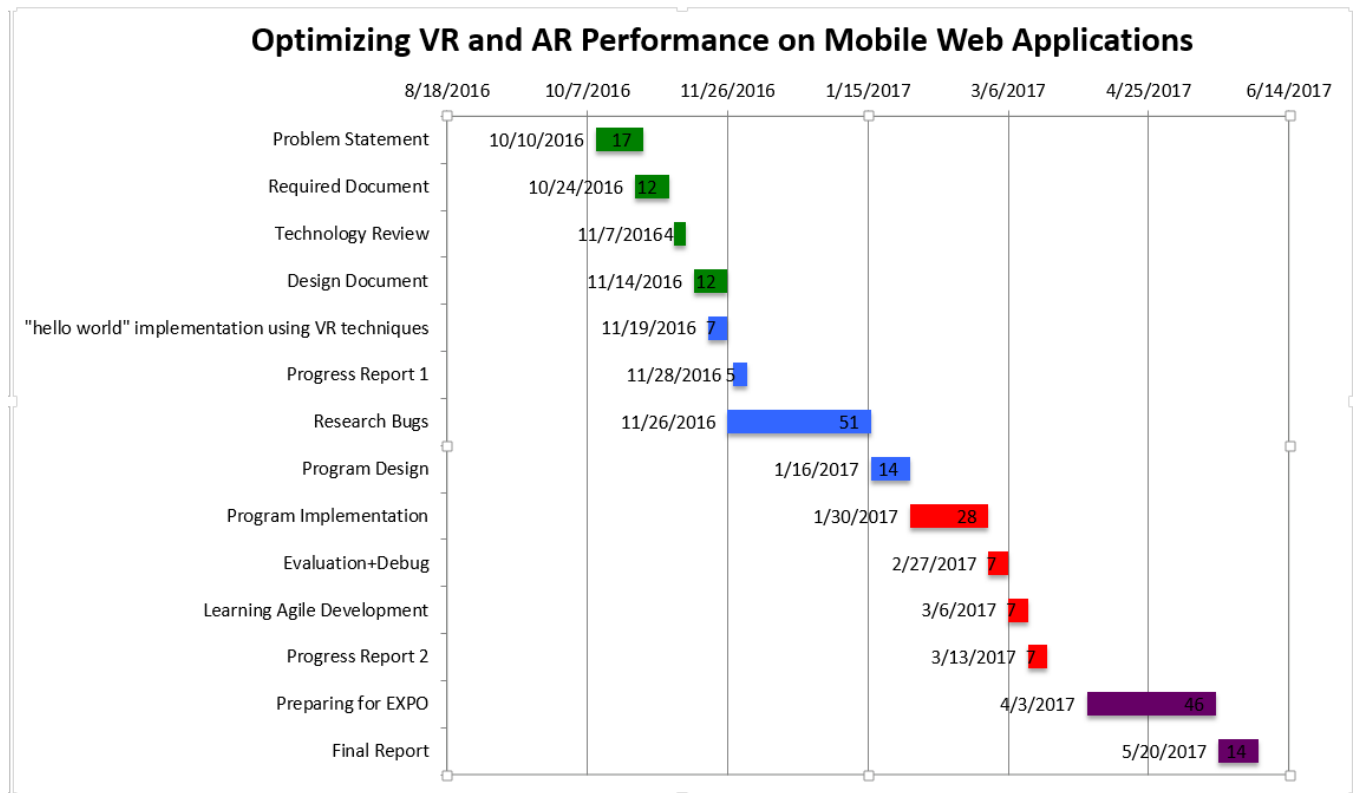
3.5.3 Maintainability

The maintainability will be based on the the dependencies of the frameworks and their function calls. This will not have a high amount of maintainability unless they do an overhaul of their base functionality. The frameworks are mostly interpreted, so if something is not rendering correctly, or if errors occur, this means that the dependencies have changed between their specific revisions. Function calls within these frameworks may change or more likely become deprecated. Since this project serves the purpose of research and testing on current versions of the software, this project will not have a large need to be maintained.

3.5.4 Portability

This software should be completely portable with little restrictions on host machines and implementation. The software we develop has to be on a web server (local or hosted) which will be accessed through a browser that interprets the markup language and the interpreted JavaScript language. There is no specific compilers necessary to run this software. There are framework dependencies that are detailed in Section 2.5, as these will often update and expand on the utilities they provide. The software that this will be tested on is Chrome and Firefox on Android Operating systems. Apple's mobile OS, iOS and their browser, Safari, will be excluded in testing and cannot guarantee it to work on that platform.

4. Gantt Chart

Optimizing VR and AR Performance on Mobile Web Applications

Task	Date	Duration
Problem Statement	10/10/2016	17
Required Document	10/24/2016	12
Technology Review	11/7/2016	4
Design Document	11/14/2016	12
"hello world" implementation using VR techniques	11/19/2016	7
Progress Report 1	11/28/2016	5
Research Bugs	11/26/2016	51
Program Design	1/16/2017	14
Program Implementation	1/30/2017	28
Evaluation+Debug	2/27/2017	7
Learning Agile Development	3/6/2017	7
Progress Report 2	3/13/2017	7
Preparing for EXPO	4/3/2017	46
Final Report	5/20/2017	14

Timeline axis: 8/18/2016, 10/7/2016, 11/26/2016, 1/15/2017, 3/6/2017, 4/25/2017, 6/14/2017

Client Signature   Date

Group Signature   Date

Group Signature   Date

Group Signature   Date