Northern Arizona University

Flagstaff, Arizona 86011

October 27th, 2016

Dr. Elmer Grubbs

NAU Faculty Member

Engineering Building (#69)

2112 S Huffer Ln

Flagstaff, Arizona 86011

Dr. Grubbs,

On behalf of the both of us, Liam Burke and John Miscevich, we would like to extend our gratitude and appreciation for you accepting us to work on your project, despite the unconventional nature of our two man team. Although we are a two man team, we are thoroughly committed to the project, and seek to overcome the disadvantageous team size through alternative tactics that only a two man team could employ. Through our alternative tactics and our hard work, we seek to produce meaningful contributions to the state of the project.

The reason that we selected this as our capstone project for Northern Arizona University’s EE476 course, taught by Dr. David Scott, was our extensive interest in the project. Our interest in the project stems not just from the actual content associated with what the project seeks out to accomplish, but also the sect of engineering that it utilizes. Performing this project would give both of us the opportunity to explore, experience, and immerse ourselves in the relatively new engineering fields of motion based user interfaces and virtual reality. We both knew that we would, undoubtedly, gain an extensive amount of skills and knowledge in a newly emerging technology, while getting to hone and further develop our already existing skills in the world of software and programming.

As an acting faculty member here at Northern Arizona University, you no doubt already understand the process of capstone projects and design. However, for the sake of comprehensiveness, we will briefly outline the purpose of this documentation: this report serves as a midterm status report for you, the client and faculty member facilitating the process. Following this cover letter, there will be a title page, a table of contents for navigation, a problem definition, research survey results, and finally, the defined requirements and specifications.

Once again, we thank you for this opportunity, and we will not disappoint you.

Sincerely,

Liam Burke and

John Miscevich



Northern Arizona University

Dr. David Scott

EE476C

Virtual Reality and Data Analysis

Midterm Client Status Report

John Miscevich

Undergraduate EE Student

jdm476@nau.edu

Liam Burke

Undergraduate EE Student

lwb32@nau.edu

November 1st, 2016

# Table of Contents

Project/Problem Definition…………………………………………………………..………...…………. .1

* High Level Project Visualization……………..………………………………………………….3

Research Survey Results Section…………………………………………………………….………...…...4

* Liam Burke Research…………………………………………………………………………….4
	+ Client Requirements and Desires
	+ Software and Hardware to be Utilized
	+ Current Technology and Potential Upgrades
* John Miscevich Research………………………………………………………………………...5
	+ Unity Engine
	+ Microsoft Kinect
	+ Oculus Rift Virtual Reality Headset
* References………………………………………………………………………………………..6

Requirements and Specifications…………………………………………...……………………...……….7

* Mechanical……………………………………………………………………………………….7
* Electrical………...…………………………………………………………………………….…7
* Environmental………....………………………………………………………………………....7
* Documentation………………………………………………………………………………...…7
* Software/GUI…………………………………………………………………………………….8
* General Requirements………………………………...…………………………………………8

## **Project/Problem Definition**

As a general overview, the aim of the project is to utilize virtual reality to view and enable a user to access and analyze large sets of data. Specifically, this project has its background in a militaristic application. Namely, the project will be utilized to view data sets and the visual representations of those data sets for different military tests. This includes tests of various aircraft, missiles, and on-land transportation. Through the use of the project, the user will be able to select iconic representations of these tests in a virtual world, select the specific test results they want to view, and then see the data in graphical form. While viewing the data in this graphical form, the user can use physical motions to analyze the data in various ways. This includes the user scaling the graph along each axis, zooming in on a certain point, selecting a data point, etc.

The benefits of this project are simplification of reading data, as well as a deeper understanding of the data for the user, in a way that is engaging for themselves and others. For instance, graphical analysis software is often difficult to utilize. By mapping certain operations to simple physical motions that anyone can do, more users will be able to view, and thus interpret, large data sets. Additionally, by having a more “hands on” connection to the data, the user is not only more engaged, but is able to analyze the data more easily, as opposed to staring at a static, flat representation of the graph on a computer screen.

Although this project is honed in on one specific application, the concept lends itself to a wide range of similar uses. For example, as the project and the technology progresses, graphical analysis via physical motion can become more normalized. With enough progression in this particular field of study, users may someday be able to view and manipulate data and graphs in a more “natural” setting. Specifically, instead of needing to utilize a virtual reality headset, a laptop, and a large “clunky” motion tracker, professionals in industry may someday be able to project data and graphs directly onto their desks, for numerous people to see, and manipulate the visualization with motion, before everyone’s very eyes. Overall, this project serves as a segue into a potential world of simple, aesthetic data analysis and representation.

To view the project on a narrower scope, this entails breaking the project into three smaller subtasks, and then coordinating those three subtasks so that they operate together as one, cohesive project.

Task one consists of developing and displaying the virtual world that the user sees. For this, the desire is that the user can sort the data through simplified visions and icons, instead of dense, uninteresting text. For instance, if a user wanted to see data sets regarding flight testing, instead of navigating to the data through text and “point-and-click” operating, they would simply be able to select a 3D image of an airplane to see such data. Task one for the team is to create the virtual world that is easy to navigate, intuitive, and contains concepts of universal usability, as well as the icons that the user sees in the Unity game engine.

Task two consists of tracking and linking the user’s physical movements to the virtual world. Although a virtual reality headset allows the user to “look around” the created world, the motion tracking affiliated with it is rather primitive. Because of this, a Microsoft Kinect will be utilized to track the user’s motion in a specific and detailed way. This motion tracking will then allow the user to now see, interact, and move within the virtual world developed in the first task.

The third task would be to implement the data aspect of the project. This specific task contains two subtasks within itself. The first of these subtasks is to visualize the data in graphical form within the virtual world. This means showing the user graphs, charts, or figures from big data sets, within the Unity Engine. The second subtask consists of developing gestures and programming the Microsoft Kinect, so that certain motions the user makes can directly alter the visual aspects of the data they are seeing. This includes zooming in or out on certain portions of the graph, scaling the graph differently, and highlighting certain points.

The culmination of these three tasks will yield a cohesive project, wherein the user can see graphical data within a virtual world, and then use his or her real life, physical motions to better select and access the data.

The following page contains a high-level visualization of the capstone project:

## **Visual System Depiction**

## Capstone_Visual.JPG

**Image 1. A Visual Overview of the Capstone Project**

Image 1 on the previous page provides a brief, high-level visualization of how the project will work. A virtual reality headset, a Microsoft Kinect, and a graphing software will all three be coordinated and constantly working in conjunction with one another to provide the user with easy-to-manipulate, interactive data visualizations that encompass the entirety of the user’s (virtual) reality. All three of the subsystems involved will be concurrently running and constantly “communicating” with each other, as these three subsystems will be co-dependent upon one another in the final implementation of the project.

## **Research Survey Results**

**Liam Burke**

The research phase began with a group meeting consisting of both team members and the project supervisor, Dr. Elmer Grubbs. This meeting provided an opportunity for both members to not only ask any questions regarding the software that will be used on the capstone project, but a somewhat casual tutorial of each of the components to the project from an individual who has been working on said project for over a year up to this point. He demonstrated several tools that would be of use in the Unity Engine, how the Kinect system would work, specifically how it could be used to read hand gestures (specific to the project) and other bodily movements, and finally, allowed each member to personally test the Oculus Rift VR headset. This allowed each member to see how the 3D environment was transferred to the interactive headset. This first step quickly allowed immersion into the software that will be used for the duration of this project. From this point, research on the Kinect, Unity Engine, and Oculus Rift was done separately.

First came a bit of research on the Oculus Rift, which dealt with reading the manual for the hardware and simply understanding how to properly use the VR headset and make sure if we ever run into a simple problem, it can be dealt with quickly and easily [6]. For the Unity Engine, there is a very helpful website which includes many different tutorials covering example projects, ranging from beginner difficulty to some more complex projects [1]. Each tutorial is fairly extensive, including not only an introduction and specific tasks to complete the given project, but general concepts to help acquaint individuals with the given software. In terms of the Kinect, besides downloading the IDE and attaining the existing source code, extensive research in regards to the functions and uses of the Kinect software has yet to be done [2]. Finally, all existing code was transferred to a personal computer so that it could be examined line-by-line to see exactly how the code operates and the functions that it possesses. Inspection of the current code is still minimal, but more shall be done in the near future, especially with the design phase coming upon us. Until now, the research that has been done has been to become familiar with the Kinect, Unity Engine, and Oculus Rift before diving into the code already written for the capstone project.

In terms of the current technology being used for this project, there are upgrades or other options to be considered. For example, the Oculus Rift in our possession is an original test model that still runs on Windows 8 rather than the most current update of Microsoft Windows. There are newer models of the Oculus Rift to look into, as well as others such as Samsung Gear VR and Google Glasses. At this point, because of monetary constraints and how much has already been completed with the current Oculus Rift, it does not seem beneficial to find a new VR headset and start from the beginning. However, this is a possibility that will be revisited in the future. In addition, the existing code was written for Unity software that has since been slightly changed and updated. As discussed with our advisor, we will be updating to the most recent version of Unity and making changes to the existing project as necessary. In terms of more creative or efficient ways of solving the problem given to us, it seems the best way to handle this project is to use the tools already given to us because of the great deal of existing source code that we already have at our disposal. If problems are encountered that cannot be solved because we are limited by the technology given to us, only at that point will we seek alternative solutions in our software and technologies.

**John Miscevich**

Due to the rather unconventional nature of the two-man capstone team, and because both team members are expected to equally and fairly contribute the same amount of work to the project, please be warned that there will be large amounts of research and information overlap between both partners, so that they may be equally as useful to completion of the capstone.

Because this project is the expansion upon a previously existing, previously worked on project, the first and foremost piece of research consisted of a meeting between both team members, as well as the Client/Faculty Advisor, Dr. Elmer Grubbs. During this meeting, Dr. Grubbs made clear the expectations, desires, technologies, and current progress on the project.

One such technology is the game development software to create the 3D virtual world. Although there is a vast array of 3D game development engines, the one that is currently in use is the Unity Engine [3]. Between the Unity Engine and the other potential game development softwares, there are many trivial, minute differences between them. However, these differences are so small, that it would serve no benefit to utilize another software and, in fact, might actually serve as a detriment to the project. The source of this detriment to the project stems mainly from the fact that there is already existing code that functions well in Unity. Utilizing C#, the Unity Engine has played a firm role in the existing project, and to change the selection of game development engines would mean not only having to entirely redevelop what has already been done, but having to port the existing C# code to different languages to accommodate the different engines (such as Lua for CryEngine, or C++ for Unreal Engine). Fortunately, the use of the Unity Engine is entirely free, and its open source nature negate any issues of patenting or market conditions. Conclusively, more research into this specific sect of the project will entail learning, examining, and developing a proficiency in the Unity Engine game development software (including the necessary C#).

The second piece of technology that is associated with this project is the Microsoft Kinect. Motion sensing and motion controlled technologies are a (somewhat) recently emerging technology, and are found in many facets of life. Perhaps most noticeably is in the world of video games. For example, Sony’s PlayStation utilizes motion sensing technology for their PlayStation Move, and the Nintendo Wii is an entirely motion based gaming console. However, a major drawback to these types of motion sensing devices is that they sense motion based on a controller, not the user’s body and physical actions. Different from the Xbox Kinect, the Microsoft Kinect (a distinction that must be made, as it is easy to confuse the two), is a motion tracking device that tracks the user’s body and body data, designed for a more intensive, “data dense” tracking. In all regards, the Microsoft Kinect best fits as the hardware to utilize for this project, and to change it would cause a large setback and delay in development. Not only would it not be feasible to purchase a new motion detection device due to budgetary restraints, but, as with the Unity Engine, it would set the project back because it require use of new software and code, independent of what has already been developed. Development for the Microsoft Kinect is done via the Microsoft Kinect SDK, and coding is done most heavily in C# and C++ [4]. Thus, extensive research into this technology will include (as with the Unity Engine, above) learning, examining, and developing a proficiency in the Kinect SDK (including the necessary C# and C++).

The third and final piece of technology that is currently affiliated with the capstone project is the virtual reality headset. Although there is a considerable (and ever growing) market for virtual reality headsets, the one currently in use is the Oculus Rift DK2 [5]. Although there are competitors in the virtual reality headset industry, all with minor variances in the hardware and software they use, the currently utilized headset is the best choice for this project. The Oculus Rift company is the most well-known, most thoroughly established VR company to date, and the DK2 is their most up-to-date version of their product. Additionally, it offers massive benefits, such as being directly integrated into the Unity Engine, so ease of development is a large advantage. However, perhaps the largest advantage of the current headset, as well as the main disadvantage of attempting to utilize a new solution would be availability. As with other components of this pre-existing project, the Oculus Rift is already in our possession. Thus, in addition to being impossible due to budgetary restrictions, purchasing a new headset would require completely redoing and redeveloping everything that has currently be done.

Conclusively, for this project, hardware (and even the choice of development tools) is an unwavering, hard constraint. This constraint produces an interesting development in the project, in that all research has taken, and will take, the form of simply analyzing code and development tools. Rather than the traditional capstone style of researching hardware, developing schematics, and potentially producing mechanical output, this project and its research will consist almost solely of developing software and programming.

References

[1] Oculus Rift. "DK2 Development Kit." DK Quick Start Guide (n.d.): n. pag. <http://static.oculus.com/sdk-downloads/documents/Oculus\_Rift\_DK2\_Instruction\_Manual.pdf>

[2] Unity. "Unity - Learn - Modules." Unity. Unity Engine, n.d. Web.<https://unity3d.com/learn/tutorials>

[3] "23 Recommended and Available 3D Game Engines." World of Level Design. N.p., n.d. Web. <http://www.worldofleveldesign.com/categories/level\_design\_tutorials/recommended-game-engines.php>.

[4] Microsoft. "Developing with Kinect for Windows." Microsoft Developer Resources. Microsoft, n.d. Web. <https://developer.microsoft.com/en-us/windows/kinect/develop>.

[5] Lamkin, Paul. "The Best VR Headsets: The Top Virtual Reality Devices to Go and Buy Now." Wareable. N.p., n.d. Web. <http://www.wareable.com/headgear/the-best-ar-and-vr-headsets>.

Additional References

[6] Oculus Rift. "Oculus Rift Development Kit 2 (DK2) | Oculus." Oculus Rift Development Kit 2 (DK2). N.p., n.d. Web. <https://www3.oculus.com/en-us/dk2/>.

## Requirements and Specifications

**Mechanical Requirements**

For this capstone project, we are limited by the wired Oculus Rift VR headset, along with the precision and accuracy of the Kinect system. Because the VR headset is wired, there is only so much distance that a user can put between themselves and the computer it is drawing power from. Along with distance, the user could potentially have issues when “circumnavigating” the virtual room, becoming entangled in the chords connecting the headset to the computer. Although this may detract from the experience, it should not hinder the overall purpose of the program, which is to allow an individual to enter a virtual room filled with different weapons and technologies, and allow them to gain information on all of the following with specifications, graphs, and more provided to the group by the advisor. Another mechanical constraint is how precise and accurate the Kinect system will be when reading the hand and body gestures of the user. Placement of the Kinect camera, along with any surrounding items potentially blocking the movements of the user, are all potential limitations for the Kinect Camera, but not necessarily a guaranteed constraint. In addition, there must always be a concern, from an organizational standpoint, as to how the user moves. Because there is a physical tether to the VR headset, the designs of the motion must be such that the user never entangles themselves in any of the physical wires coming from the hardware. Finally, whenever not in use, the hardware must be adequately and properly stored in an indoor environment, as to prevent degradation to the hardware.

**Electrical Requirements**

The setup for this project will be as follows: the Oculus Rift VR headset will be connected to a personal computer, which will be running the Unity program we devise, and the Kinect camera will also be connected to the computer and facing the user. After finding that the Oculus Rift draws 5V of power while the Kinect typically draws about 12V, power consumption will not be a problem for this design. However, aging is a constraint that the group must take into account. Our aging technology may not only become obsolete because of the ever changing and updating devices, but because over time, the VR headset and Kinect Camera may endure some wear and tear and not always be as efficient and reliable as when they were first used. One thing to keep in mind during the design and implementation of the project is to try and ensure the longevity as long as possible. This entails making the equipment and software as “universal” as possible, so that new releases of hardware or software will not instantaneously deem the project obsolete or non-functioning.

**Environmental Requirements**

The hardware and software will, as of now, solely be subjected to conditions faced in an indoor environment, so it must be able to withstand these relatively tame conditions, both in use and in storage.

**Documentation Requirements**

The separate pieces of hardware and software (Oculus Rift, Kinect, and Unity Engine) each have their own User’s guides and Operator Manuals. This project will need to include its own user manual that shows how to setup the hardware used for this project, along with instructions on how a user can navigate the virtual 3D environment that we will be creating. Within the manual, there shall include guides on how the user can choose specific elements in the environment, how they can maneuver their way through different types of data (specifications sheets, data sheets, graphs, etc.), the hand gestures that can be used with the Kinect camera, and more.

* User Guide: This project will have a user guide that includes how to setup the hardware provided with the design, along with how an individual would go about using the system developed in Unity.
* Maintenance Manual: A manual will need to be incorporated that will allow a user to provide maintenance to the system without having to receive professional help.

**Software/GUI Requirements**

The group is limited by the fact that because the Oculus Rift VR headset is an original prototype, computers with software updates beyond Windows 8 cannot be used in the process of completing this capstone project. This means that work for this project will have to be completed solely on the computer provided to us by our project supervisor. In addition, the group will be using the Unity Engine and Kinect IDE, as described by the project supervisor, and constraints will become more apparent as the project develops into further stages. As of this moment, it would appear that we are limited to what the Unity library offers us in terms of three dimensional figures to be placed in our virtual environment. We have decided, in terms of the Kinect, that we not only want to incorporate hand movements to this project for choosing elements in the virtual environment, but incorporate voice commands to the system using the Kinect.

* Unity Engine: Unity will be the software used to create the 3D virtual environment.
* Kinect: The Kinect will be used to allow the user to interface with the system using hand gestures, body movements, and voice commands.
* Oculus Rift: The Oculus Rift is the VR headset that will be used, allowing the user to have a more realistic connection to the 3D virtual world.
* Windows 8 OS: As a result of the Oculus Rift software, the computer used in the project must be using Windows 8 Operating System.

**General Requirements**

This project needs to have longevity, and be able to be easily updated and modified so that when new technologies are available for use, this project will not become completely obsolete. This project must also be easily understood and operated; for the user to get the most out of their experience with the program, it must be absolutely clear what can be done in the program, along with the ways the tasks can be accomplished. Finally, the experience for the user must be a comfortable one; an individual would not be able to enjoy a 3D virtual environment if they are not comfortable while working with the program. Overall, the general requirements are the desires of the client and advisor, Dr. Elmer Grubbs. Additionally, because the project is heavily software and user oriented, the concerns of the user must always be kept in mind. This includes making the project as intuitive, easy to learn, and comfortable as possible by utilizing principles of universal usability.

**Conclusion**

Due to the nature of the project, there are not too many strict, quantitative specifications. However, the entire project is heavily design oriented, and must be developed with various user interface, user experience, and design considerations.