Northern Arizona University

Flagstaff, Arizona 86011

March 10th, 2017

Dr. Elmer Grubbs

NAU Faculty Member

Engineering Building (#69)

2112 S Huffer Ln

Flagstaff, Arizona 86011

Dr. Grubbs,

On behalf of the Virtual Reality Capstone Team, John Miscevich and Liam Burke, we would like to invite you to the Capstone Design Conference on April 28th, hosted by Northern Arizona University’s College of Engineering. Should you be interested in attending, our team will provide you with more information as the presentation date approaches.

As of this point in the semester, many aspects of the project have remained unchanged. For instance, the design requirements, specifications, and deliverables are still the same as previously determined during the planning phase of the project, as well as the proposal. However, one change has occurred in regards to the budget. Specifically, one purchase was necessary to implement the graphing abilities of the project fully and completely. This purchase of one Unity Store asset, Graph Maker, cost forty-five dollars, USD, and is described in further detail in the “Capstone Budget” portion of the report.

Overall, a great deal of progress has been made in regards to the project. Some key tasks that have been accomplished include integration of motion controls into the virtual environment, navigation between selection menus, and the ability to visualize graphical data. This report seeks to explore these concepts in greater depth, as well as discuss any changes and difficulties that have arisen as the project has progressed. Additionally, this report contains a new addition that has not previously been present in prior reports – a Comprehensive Test Plan section. This section will explore the completed tasks, in depth, based on the testing that has been done to complete those tasks. Additionally, it includes testing that will be done in the future for tasks that are on the timeline to be completed.

In regards to the project’s timeline, virtually everything is on course and going smoothly. However, there exists one task that is taking a little longer than anticipated – the manipulation of the graphs via motion control. Currently, the project contains both individual aspects necessary to control graphical representations. The project is able to display graphical data in a way that is clean and aesthetic, and the program is able to track the user’s body motions to perform selected tasks. The issue at hand, then, is combining these two aspects together to produce the desired outcome of altering the graphs via motion controls. Our team is only a day or so behind schedule, but should resolve the issue soon. Additionally, our team is not too concerned with this slight delay, as we had predicted that this would be the most time consuming aspect of the project. Other than this singular issue, all other tasks are on time, to be completed as expected.

So that you may always stay informed on the status and state of the project, we ask that you occasionally visit our website at:

https://cefns.nau.edu/capstone/projects/EE/2017/VirtReality/index.html

This website contains the project overview, description, system depiction, a team biography page, and a page that outlines the four stages of engineering design (and how they pertain to our project).

One request that we ask of you is that you visit our website at your earliest convenience and provide us with an informal critique. We would like some recommendations on items such as the website’s layout, color scheme, content that can potentially be removed, and content that you would like to see added. Given these recommendations, we will make the necessary changes, so that the project may be adequately and fully represented. Thank you.

Sincerely,

Liam Burke and

John Miscevich



Northern Arizona University

Virtual Reality and Data Analysis

Client Status Update

March 10th, 2017

John Miscevich

Undergraduate EE Student

jdm476@nau.edu

Liam Burke

Undergraduate EE Student

lwb32@nau.edu

Table of Contents

Executive Summary …..……………………………………………………………………………………1

Problem Definition and Project Overview ...……………………………………………………………….3

System Depiction …………………………………………………………………………………………4

Research ……………………………………………………………………………………………………5

* Recently Conducted Research (Spring 2017 Semester)……………………... …………………...5
* Previously Conducted Research (Client Proposal, Fall 2016 Semester)……… ……………….....7
* References………………………………………………………………………………………….9

Requirements and Specifications………………………………………………………………………….11

* Mechanical, Electrical, and Environmental Requirements and Specifications ………………….11
* Documentation, Software/GUI, and General Requirements and Specifications ………………...12

Project Design……………………………………………………………………………………………13

Capstone Budget …………………………………………………………………………………………14

Comprehensive Test Plan…………………………………………………………………………………15

* Completed Tests………………………………………………………………………………….15
* Future Tests …………………………………………………………………………………….16

Schedule and Deliverables ………………………………………………………………………………..18

* Key Milestones and Tasks Completed………………………………………………………….18
* Remaining Tasks ……………………………………………………………………………...18
* Deadlines and Deliverables……………………………………………………………………..19

Previous Documentation…………………………………………………………………………………..19

Appendix A – Gantt Chart ………………………………………………………………………………..20

**Executive Summary**

This executive summary seeks to expand upon topics that are critical to the project, including a summation of the project, completed work, proposed design concepts, deliverables, and budgetary information. Although bearing many similarities to the Client Proposal deliverable that was presented at the end of the Fall 2017 semester, this report aims to serve as a continuation of that proposal, including any changes in the project, as well as an outline of work that has been completed and work that is still in progress.

Fortunately, many critical aspects of the project have remained unchanged, thus far. For the sake of this client status report, these components will be revisited and included in this documentation, under their respective headings. The project components that have remain unchanged include the problem definition, the system depiction, the requirements and specifications, and the design. Because there have been no modifications to these portions of the project, their inclusion in this status report will be nearly, if not completely, identical to their corresponding sections in the Client Proposal deliverable. There has, however, been additions and modifications to approximately four project components: the research, the budget, a comprehensive testing plan, and the schedule. To explore the changes that have been made to these three aspects, please view their respective sections of this report. All sections may be found and referenced via the Table of Contents on the previous page.

Thus far, many key milestones of the project have been completed and a great deal of progress has been made. For example, all of the virtual environments are completed in their entirety. Beginning on the equipment selection scene, to the model and type selection scene, to the date selection scene, and finally to the graphical data scene, all of the virtual “rooms” are created. Additionally, motion controls have been fully implemented into these scenes, meaning that the user is able to see their hands moving about the physical space, responding to stimuli in the virtual rooms, and activating menu items. Once the menu items have been activated, they respond accordingly (such as loading the next scene). Finally, graphs are now viewable in the virtual world. Taking in a desired data set, information can be displayed in varying graphical formats, including pie graphs, line graphs, bar charts, etc. These graphs are titled, labeled, and easy to read.

Despite the overall progress, there still exists work that needs to be completed. Specifically, the largest task to-date has been the integration of motion controls into the graphs to create an aspect of interactivity. Although motion controls, by themselves, are implemented, as well as the graphical representation of data, these two components have not yet been merged into one functional component. This is currently the one delay in the project’s timeline, but our team is not concerned, as we had anticipated this to be the largest portion of the project and have, thus, allotted ourselves a good spacing of time to complete the task. Additionally, to amend this slightly missed deadline, the team will expend a great deal of time and effort in catching the project up to speed. Once the motion controls and the graphical representations are integrated together, the project will be mostly complete. At this point, the team will begin work on importing data into the project, developing the mobile application, and producing final deliverables.

Currently, the team and the project maintain a public web presence via the domain provided by Northern Arizona University:

https://cefns.nau.edu/capstone/projects/EE/2017/VirtReality/index.html.

This webpage contains a great deal of pertinent information about the project and can be utilized by both the sponsor, as well as the public, to get a glimpse into the current happenings and recent developments. It is a phenomenal tool that will keep the site’s visitor informed and up to date on the project’s status.

Overall, the project is going well and the team is satisfied with their current progress. However, there is still a good deal of work that needs to be done as the Spring 2017 semester enters its second half. The team hopes to enter the second half of the semester in full swing and continue making meaningful progress on the project, with the overall aim of fully completing it to Dr. Grubbs’s desires by the end of the semester.

#

**Problem Definition and Project Overview**

Presently, the project definition has not been modified in any way and is, thus, unchanged. Thankfully, the knowledge and information that has been gained as the project has progressed has not led to the necessity to redefine the problem statement. For this reason, the below Project Definition is the same one that can be found in previous documentation and deliverables.

Presently, the project overview and system depiction has remained unchanged. Fortunately, the knowledge and discoveries gained through research and experimentation have not led to the necessity to restructure the project in any way. In fact, the project has been progressing quite smoothly as it was originally planned, and thus the original project overview and the original system depiction still serve to be accurate and complete.

As a general overview, the aim of the project is to utilize virtual reality to view and analyze large sets of data.

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 world and the icons that the user sees in the Unity game engine, and provide it functionality.

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 them is rather primitive. Because of this, a Microsoft Kinect will be utilize to track the user’s motion in an immensely 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 contents 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 modify the visual representations.

**System Depiction**

As with the Problem Definition, the System Depiction also remains unchanged. For this reason, the below System Depiction is the same one that can be found in previous documentation and deliverables.

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

The image on the preceding 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**

As a year-long, continuous capstone, a great deal of overall research has been conducted in regards to this project. Thus, this Research section of the report begins with the topics that have been researched most recently, during the duration of the Spring 2017 semester, from January 15th to February 24th. However, this section also includes all previously conducted research, as included in the Client Proposal.

**Hardware Necessary to Run Virtual Reality:**

One of the previously referenced “project issues” was the inability to run the required virtual reality on the laptop provided by Dr. Grubbs and Northern Arizona University. Thus, the team spent a good deal of time trying to develop a solution, including researching viable graphics cards and laptops that would be capable of running the necessary software. During the research phase, it was discovered that the provided laptop was not compatible with any existing graphics cards that are able to support the Oculus Rift Development Kit 2. Thus, it was decided that the only solution to provide an aspect of portability to the project was to purchase a new laptop. Although a laptop is not completely necessary for the project (there is a computer in the EE lab that is capable of running the laptop), it is preferred by Dr. Grubbs, as it would give the project an aspect of portability. Dr. Grubbs identified one laptop that would be a potential candidate, the Dell Inspiron 15 i7559 series. After conducting the necessary research, it was discovered that this laptop would, in fact, fully support the Oculus Rift DK2. Whether this laptop will be purchased or not is still at the discretion and decision of Dr. Grubbs.

**How to Graph in Unity**

With much of the menus, navigation, and motion controls integrated into the project, the next phase of research was in regards to how to display the desired graphs in Unity. One outlet that was explored was utilizing particle systems to present graphical data. This seemed to be a promising solution, so it was explored further. By entering data into the project, each data point would individually be placed into the virtual world as a single particle. All of these particles, then, would collectively act as the data representation (be it a line, a parabola, an abstract scatter, etc.) The following image shows what this particle system representation would look like:



**Image 2. Some Examples of Particle System Graphing**

The idea of particle system graphing had numerous benefits to it. For example, it is a relatively simple, neat way to graph various 2D and 3D functions, all within the confines of Unity’s engine (meaning no external program would have to be referenced). Additionally, the resulting depictions are interesting and keep the user engaged. However, the concept did raise a few concerns and questions. For example, how would axes be implemented, how would data points be labeled, and how could the graphs be manipulated, among other questions. Because of this growing number of concerns, more potential outlets of graphing were researched.

Upon extensive research, the utilized solution was a Unity asset. Unity assets are pre-built tools provided by Unity developers and users to accomplish commonly needed tasks. In this particular instance, there exists numerous assets that accomplish the task of creating clean, easy-to-read graphs. Research was done to determine which graphing asset would best accomplish the goals of the project, and it was decided that GraphMaker was the best choice. This asset was then purchased, a detail that will be explored further in the “Capstone Budget” portion of the report.

**Mobile Application Development**

As the second half of the final semester is approaching, it has become necessary to begin preliminary research on some of the later aspects of the project. This includes the development of a mobile application that can be used in parallel with the virtual reality project to produce data that can then be graphed. This portion of research has been relatively brief, but it has been insightful as to how to produce mobile applications for Android phones. It includes reading Android Studio documentation, as well as practicing and becoming familiar with the Android Studio desktop application. Upon completion of the project, it is the hope of the team that there is an accompanying mobile app that will track some form of data (such as accelerometer data or GPS location data) and then display it in one of the graphs in the virtual environment. The representation of the data, then, will be able to be manipulated just like all other graphs within the program – via the user’s physical motions.

**Previously Conducted Research**

**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. [1] For the Unity Engine, there is a very helpful website, https://unity3d.com/learn/tutorials, which includes many different tutorials covering example projects, ranging from beginner difficulty to some more complex projects. 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. [2] 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. 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/>.

[7] "Graphs, a Unity C# Tutorial." Graphs, a Unity C# Tutorial. N.p., n.d. Web. Nov. 2016. <http://catlikecoding.com/unity/tutorials/graphs/>.

[8] Textcube. "Unity 3D Box Chart Graph Quick Making." YouTube. YouTube, 07 July 2013. Web. Nov. 2016. <https://www.youtube.com/watch?v=ilwtKe4CFIU>.

**Requirements and Specifications**

As with the project overview and the project depiction, the information gained via research has not led to any major necessity for restructuring requirements and specifications, and fortunately, the project has been progressing smoothly enough that no additional, hindering restraints have emerged. Because of this, the below requirements are identical to those found in the finalized Client Proposal deliverable, provided at the close of the Fall 2017 semester.

**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 tangles 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.

**Software/GUI Requirements**

One major limitation of the project, currently, is the large graphics requirement that is associated with virtual reality. Namely, to keep the project portable and runnable on the given laptop, the laptop is constrained to the Windows 8 Operating System. Unity and Oculus are capable of running and, in fact, even run better on Windows 10. However, the laptop provided does not have the proper graphics card to run the necessary softwares on Windows 10. In addition, the group will be using the Unity Engine and Kinect SDK, 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.

**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.

**Project Design**

During this phase of the capstone project, no new design decisions have been made. Rather, the project has simply consisted of implementing and executing the previously planned designs. For this reason, the Project Design segment of this report remains unchanged from the previously submitted Client Proposal.

**Design Description**

Many of the chosen design concepts and approaches are those that were provided by Dr. Grubbs, himself. For example, Dr. Grubbs has recommended that the Unity 3D game engine be used for designing the virtual world and the Microsoft Kinect SDK be used for tracking and logging motion. Through various research and experimentation, both of these suggestions have been proven to be the most viable options, and will thus be implemented in the project.

Additionally, the layout of the implemented project has been firmly set as a desired constraint, from the beginning. Namely, Dr. Grubbs, the client, has made it clear how he would like the user experience to progress. Specifically, he would like the users to begin by seeing iconic representations of military equipment, such as a tank, an airplane, and a missile. Upon the user selecting their desired equipment, they will be prompted for more information, such as the specific model of the equipment they selected, as well as the type of testing that was done. Once the user has selected these two parameters they will, of course, be taken to the scene that will display all of the testing done that meets those requirements. Once on this scene, the user can finally select from individual tests that were done, each of which will take the user to the graphical data, to be visualized and analyzed at their leisure.

Because these are firm constraints set by the client, the design mentioned above will be utilized and implemented. However, there may exist some variation in the design of each individual scene, in terms of user interactivity and aesthetics. These design decisions, at the time of their implementation, will be presented to Dr. Grubbs, approved, and then implemented. Upon completion of each scene, Dr. Grubbs will experiment and prototype the existing project and make clear his satisfaction, or state improvements he would like to be made.

Due to the nature of this project, the design of each scene is very much tentative and can change per the client’s request. The client has made clear his desires in terms of what he wants and how he wants the project to function, so it is simply a matter of the capstone team implementing his wishes and ensuring that they meet his expectations, after they have been completed. Any time that the client’s expectations are met, the scenes and functionalities will be saved and improved upon for the final project. Any time that the client feels his expectations or his vision for the project are not congruent with what is delivered to him, the capstone team will revisit that portion of the project to make improvements upon it.

**Capstone Budget**

This particular segment deserves special attention, as it is completely different from previous deliverables. Whereas previous budgetary reports had indicated that there were no purchases made, nor would any purchases be necessary, this report will include some modifications to that statement. Namely, a purchase has been made out of necessity for the project. Please see below, a bill of materials, accompanied by a justification of the purchase made.

**Table 1. Virtual Reality Capstone Bill of Materials**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Item** | **Vendor** | **Publisher** | **Cost** | **Status** | **Paid By** | **Reimbursed** |
| Graph Maker Unity Asset | Unity Asset Store | Stuart S | $45 USD | Received | John Miscevich | Awaiting |

**Justification**

Although brief, the above bill of materials represents all items currently purchased for the virtual reality capstone project. The singular purchase was thatof a Unity asset – a pre-built tool made by independent Unity developers and coders for the purpose of selling to other users that require a commonly needed task to be performed. This particular Unity asset is called Graph Maker, an asset sold by Unity, from publisher Stuart S. It is capable of displaying various different types of graphs in a clean, easy-to-read fashion. This asset, of course, bears its necessity in that a great deal of the project involves displaying graphical data in a virtual environment. The cost of this asset was forty-five dollars USD and was paid via team member John Miscevich’s credit card (with the approval of both Dr. Scott and Dr. Grubbs). The purchase has not yet been reimbursed, but all of the necessary paperwork has been filled out and the proper procedure has been followed. As it is a digital item, it was instantly received upon payment, in the form of a download.

Not only does the above bill of materials reflect the items that have already been purchased, but it reflects the additional items that will need to be purchased. Specifically, the team predicts that no further purchases will be necessary for the completion of this project. Thus, the total and final cost of this project is predicted to be forty-five dollars USD. However, should an additional necessary purchase arise, as always, the team will utilize the proper outlets to go about making the needed purchase.

**Comprehensive Test Plan**

Because this is an almost exclusively programming and software based capstone project, we ask that special consideration be given when examining the testing utilized to ensure the project is working. Due to the nature of the project, testing is almost constant. Whereas other projects may require a good amount of effort to set up a test, run it, and make note of the results, testing of a software based program can be done instantaneously. A small amount of code may be written before it is compiled and “tested” to see if it runs properly, all in the span of mere minutes. Because of the simple and instantaneous nature of being able to test and debug the code, the result is that the program may be “tested” hundreds of times a day. For the sake of this report, the dozens to hundreds of compilations of the code that are tested each day have been clumped together, based on desired outcome of the testing. This provides a concise, neat way to summarize the numerous daily “micro-tests”. The title of the test category is provided, followed by the team member responsible and a concise summary of the purpose of the testing. Although these tests may seem informal, the nature of the project dictates that this is how testing is conducted, for the most part.

**Completed Tests**

**Selection Scene Visuals – John Miscevich**

The Selection Scene is the first part of the project that a user will see. During this scene, the user selects either the missile, the tank, or the plane. This first group of testing consists solely of developing the visual layout of this first virtual environment. This large grouping of tests consists of over hundreds of sub-tests whose sole purpose is to see if the environment looks “right” to the user. In fact, most of these tests tend to last between 5 and 20 seconds. To perform this test, one or both team members (depending on who is available and who is responsible for testing) will compile and run the code while wearing the Oculus Rift DK2. They will then look at all of the aspects of the virtual environment and make note of what to change. Because virtual reality is so heavily influenced by visuals and perspectives, this is an incredibly critical point. Some examples may include: noting that the table the items are set on is too large, the lighting on the items is not adequate, a particular item looked better as one color than another, the items are not rotating properly, and many other visual considerations. Once something that is visually displeasing is noted, the code is modified and then retested. This process repeats itself indefinitely, until the virtual environment has achieved the desired look.

**Model and Test Scene Visuals – Liam Burke**

As with the previous selection scene, hundreds of “micro-tests” are done to ensure that the virtual environment looks in such a way that it can be easily interacted with by the users (nothing is too confusing, everything is intuitive to use, nothing makes the user nauseous, etc.). Hence, much of the testing consisted of placing a menu item into the virtual environment, running the code while wearing the Oculus DK2, and analyzing the surrounding area. If an item that was placed into the world was aesthetically pleasing, it was kept in its position and new components were added. If an item was not displaying in a desirable way, the code was changed and retested. Once again, this process was repeated until the environment looked as desired.

**Date Selection Scene Visuals – Liam Burke**

Once the Model and Test Scene had been created, developing the Date Selection Scene was simply a matter of using the Model and Test Scene as a template. The same items from the Model and Test Scene were recycled, renamed, or slightly repositioned to create the Date Selection Scene. However, the Date Selection Scene did undergo the same testing process as the previous two scenes. A vigorous cycle of “write code, compile, analyze visuals, repeat” was utilized until everything looked as it should.

**Graph Scene Visuals – John Miscevich**

In a similar fashion to all other visuals testing, testing the Graph Scene consisted of dozens, if not hundreds, of recompiles and live tests. Some tests include repositioning the camera, repositioning the graphical data, and modifying the canvas that the graph was displayed on. Once again, there was a constant, continuous cycle of recoding and retesting until the Graph Scene displayed how both team members wanted it.

**Motion Controls Testing – John Miscevich**

Once all of the scenes had been created, the next phase of the project was to begin integrating motion controls via the Microsoft Kinect. This was an extensive trial and error process of coding for the Microsoft Kinect, including it in Unity, and seeing how the project would respond. To adequately test this, two small spheres were placed within the virtual world. C# scripts were then coded and attached to the spheres to track the motion of the user’s two hands. Among the numerous constant recompilations, some of the more common errors that needed to be addressed included: repositioning the spheres in the virtual world, the Kinect not properly tracking the user’s hands, debugging problems in the code, resolving syntax errors, and recoloring and resizing the spheres. These numerous sub-tests were all accomplished with the purpose of being able to track the user’s physical movements within the virtual world.

**Future Tests**

**Motion Tracking and Graph Integration (March 15th)– John Miscevich**

Perhaps one of the most substantive aspects of the project, the ability to use physical motions to manipulate graphical representations of data is absolutely critical, and will comprise of a majority of the project’s testing. As with all testing in a software based project, the sub tests of the integration testing will be a continuous trial and error cycle of coding, running, and recoding. Testing will be done by coding various physical motions and manipulating the graph to match those motions. Some criteria to examine while testing will be: how intuitive is the physical motion, does the motion make sense paired with the action it is supposed to perform, does the physical motion alter the graph in the way that it is supposed to, is the gesture easily detectable to the Microsoft Kinect, among many more considerations. Once this vigorous cycle of testing is complete, the project should fully be able to perform its desired function of modifying graphs via physical gestures.

**Data Import Testing (March 25th) – Liam Burke**

Once the graphs can be modified via physical controls, one of the more final steps of the project will be to ensure that any desired data set can be imported into the project. Without the ability to import custom data, the project will serve as little to no use. The methodology for this is still being determined, so the testing method can not be fully described just yet. However, there exists a general outline as to the procedure that the testing will take: the team will input their desired data set, run the program, and ensure that the correct data is being visualized via the graph. In the event that the correct data does not appear on the screen, the team will troubleshoot the code and try again. This method will continue until the user can input any data they want into the project and be able to visualize it in a way that is clean and easy to read.

**Final, Full Scale Testing (April) – John Miscevich**

This final stage of testing will be the full implementation of the entire project. It will be a top down run of the project, beginning from the very first scene all the way through to the end. During this stage, both team members will essentially try to make the program crash in any way possible, so that they may resolve the issues they find. Any time a new problem is discovered, it will be dealt with and retested. If no issues are found, it will be deemed a successful run through. This stage of testing may include having third party testers, so there is an outside perspective as to the layout and functionality of the program. The third party users will then be prompted as to their general opinion of the program and possible critiques. The team will then take these suggestions into consideration and decide whether or not to implement them or not.

**Conclusion**

Our team understands that these testing specifications are broad and informally defined. However, due to the heavily software oriented nature of the project, this becomes a necessity. Additionally, these categories of testing are tentative and may change as the project evolves. All tests that are conducted will be well documented and included in future deliverables, as well as on the project’s website.

**Scheduling and Deliverables**

Because the project is mostly on schedule, a great deal of the Scheduling and Deliverables portion is similar to that of the schedule provided on the Client Proposal Deliverable. However, there are some slight modifications that have been added.

**Key Milestones and Tasks Accomplished**

* Full visualization of the Selection Scene
* Full visualization of the Model and Test Type Scene
* Full visualization of the Date Selection Scene
* Full visualization of the Graph Scene
* Full menu navigation between scenes
* Full motion integration on non-graphing scenes

**Tasks Behind Schedule**

* Motion and graphing integration

To verbally explain the above lists, a great deal of milestones have thus been reached. Essentially, the entire virtual environment has been created. All 4 necessary scenes have been created in their entirety and represent a clean layout that aims to be intuitive for the user. Additionally, the Microsoft Kinect motion controls have been integrated to some extent in the project. Specifically, the user’s hands can be tracked and monitored, interact with other project components, and trigger actions.

However, the one task that has slipped a small amount behind schedule is the integration of motion controls into the graphing aspect of the project. As of now, this has fallen short of the deadline by mere days. Despite this delay, the team is not incredibly concerned, as this portion of the project was predicted to be the most arduous and time consuming and, thus, extra time was allotted for it. To make up for falling behind, the team will expend a great deal of time and effort to ensure they are caught up within the next week, taking advantage of the time given during the spring break.

**Remaining Tasks**

* Motion and graphing integration (as described above)
* Importing custom data sets
* Mobile application development
* Overview and touch ups
* Create user manual

To explore the above list, although much has been completed in regards to the project, there still remains a great deal of work to do. As described in the above section, one of the larger tasks to accomplish is the integration of motion controls into the graphing portion of the project. Once the project is able to alter graphs via physical motion controls, the team will begin working on how to fully import their own data sets into the program for manipulation. Once that is finalized, the team will begin developing a mobile application to collect some form of data (potentially accelerometer or GPS location data) and import that data into the project. Once these components have been completed, the project will be essentially finished. The only task that will follow will be touching up the project and developing a user manual, so that it can be utilized without the assistance of either team member. All of these remaining tasks will be conducted by both members of the team in a joint manner. However, the team member that will be held most responsible for each task is the member that is cited in the Comprehensive Test Plan section of the report, above.

**Deadlines and Deliverables**

* Completed user manual (April 24th, 20147)
* Capstone design conference (April 28th, 2017)
* Zip folder containing the code for the completed project (May 5th, 2017)
* Mobile application and associated code (May 5th, 2017)
* All hardware returned to Dr. Elmer Grubbs (May 5th, 2017)

As the above list indicates, the actual list of deliverables is relatively small. The only items that will be given to the client, Dr. Grubbs, will be the code containing the fully functioning project, the mobile application, and the user manual. These deliverables will be given near the close of the Spring 2017 semester, along with the hardware that was loaned to the team.

The following page contains **Appendix A**, which displays an updated Gantt chart. Because the project is relatively on its proper timeline, the Gantt chart displayed in Appendix A bears an almost identical resemblance to that Gantt chart provided in the client proposal. However, because there was one deadline that was slipped, the chart has been amended to allot for extra time to complete the motion tracking and graphical integration task. Other than this one component, all tasks and deadlines are expected to be completed in the time that was originally set.

**Previous Documentation**

The report calls for previous documentation, electrical schematics, analysis, test data, or drawings to be provided in this section. However, because no such documentation exists for this project, it will not be included. For previous documentation such as proposals, minutes, or activity reports, please ask a member of the active team.

**Appendix A**



**The Updated, Tentative Gantt Chart of Various Milestones and Deliverables**