CS Capstone Requirements Document

May 16, 2020

Augmented Reality Collaboration

Prepared for

Oregon State University Instructor

Kevin D McGrath

Signature Date

Prepared by

Group CS 13

ARC

Carson Pemble

Signature Date

Ryan Miura

Signature Date

Haozhe Li

Signature Date

Abstract

Our client faces the problem of collaboratively annotating PCB boards from remote locations. There is currently no technology to facilitate this at a reasonable price, so we propose using a Zed Mini camera attached to an HTC Vive headset and our personally developed software to allow this. Our software will allow users to collaborate on board designs and view board physicality independent of one another while sharing visuals, audio, and user-added virtual markups. We will measure our progress through iterative prototypes, allowing our client to test our work. This will result in saving the Oregon State University research department a lot of money, advance affordable AR (Augmented Reality) technology, and help our client further his research.

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to create an agreement between the project team, ARC, and our client, Kevin McGrath, for what will be completed during the time frame of this year. This will cover the basics of what our client will be receiving once our product has been completed. Throughout this paper there will be a layout of what steps will be completed and with the inclusion of a Gantt chart, when those steps will be completed.

1.2 Product Scope

The scope of this product is to construct and develop a product that can will be usable and helpful for our client's research and ease of collaboration with colleagues. The time frame is one year, ten weeks of research and documentation, fifteen weeks of development, and five weeks of finalization and presentation.

2 Overall Description

2.1 Product Goal

The overall goal of our product is to have 2 or more AR headsets that are able to share visuals, audio, and virtual annotations with low latency. We would like to have multiple headsets, preferably three, connected through a LAN network with AR working at a single location. Everyone wearing a headset should be able to see what the AR headset person is viewing and everyone should be able to create virtual markups/annotations. The users should be able to switch cameras to create a "shared screen" experience for the other two users. This should all be done in real-time, meaning a live feed from one headset to the other, and this will also need to be done securely and safely. Once the product is complete, Mr. McGrath and his colleagues will have a foundation for a trustworthy and simple collaboration suite where they can simply put on a headset and share their work across the network.

2.2 Operating Environment

As our client has requested the software will be developed on either Windows systems and will tested using the computers in CGEL, Batchellor Hall at Oregon State University. The code will developed in Unity 3D and it will use SDKs from both of the hardware companies, HTC and Stereolabs. HTC Vive and Zed Mini have a code base that will also be used in our development process, with the help of Unity Plugins.

2.3 User Documentation

Documentation is required as a key element to this class and it will be created throughout this term and updated throughout the development process. Developed code will contain comments to explain our work and the logic of our programs. Presentations will be given at the OSU Engineering Expo and to our client directly.

3 Hardware

3.1 ZED Mini

As with most computer science projects, there is still a large need for hardware. Through the development of this project, we are required to create a few key pieces of hardware. We have been requested by our client to create an AR headset from two pre-existing products: HTC Vive and ZED Mini.

One of those is a 3D camera from Stereolabs called the ZED Mini. This camera allows for stereo pass-through which will be used to help us create an AR experience.

images/ZED.jpg

3.2 HTC Vive

Another piece of hardware that our client will be providing for us to use is the HTC Vive. This is a VR headset that the ZED Mini will be attached to the front of, to simulate an experience as if the user was looking through the headset. This newly created AR headset will be connected to a PC to power the graphics and two hand-held controllers which will allow the users to rotate and mark up the Virtual PCB board which are the key required features of this project. All hardware construction/combining will be done very early on in our development process and will be completed as a team.

images/HTC.jpeg

4 Augmented Reality

4.1 Visuals

The required project before us is to create a product that will overlay virtual items onto what the ZED Mini streams as reality. This means there is not a full virtual reality created but instead an augmentation of reality. This will be done through collecting visuals and audio and then overlaying virtual items on top of those.

The Unity3D software we use for development is able to work with the Zed Mini camera, and it supports AR markups on the camera feed. Development can start with getting video streaming to work from the camera to the headset screen. Connecting the Zed Mini to a single headset and displaying the camera feed to the user provides an indication of success. From here, virtual markups should come into the visuals, allowing for writing virtual text on the display. This text does not have to be scaled or positioned correctly at the first development, but will have to work by the final prototype for the expo. Being able to draw to the screen and view it is the next required step to our project. Then, scaling and positioning will become a focal point. Using ArUco images, the markups will be scaled to accurately indicate the components they are referencing. These marker images would be used to indicate which side of the board is currently being looked at and to reference the size of the object. Since the markers will have a known size, they can be used as constants. Once these have all been implemented into our project, we will have reached all the visual requirements. There are more requirements such as audio and more defined virtual markings.

4.2 Audio

Another key element to our project is that our software must support audio communication, as we are trying to build a collaboration product, audio communication is very important for proper communication. The HTC Vive has headphones built- in, which could be used if the user does not want to use personal headphones. There is software within the headsets to allow for audio to be passed from the PC to the headsets. This software may work for this purpose, but it may also require manual construction of an audio connection. It is also important the audio is not lagging behind the video or collaboration could be a struggle. Because of this, we are aiming to have an audio latency of less than one second for this project. Our hopes are that we can cut this down even farther. Regardless, creating an audio connection is key to this project's success.

4.3 Virtual Markings

One of the key features that our client is requesting is that all users should have the ability to circle or annotate specific components in the PCB in the virtual space. The first step in implementing this feature is to first start by being able to add some sort of visuals to the screen at the user's command. The user will be able to use their controller to point somewhere and create a virtual marking, like a circle on the PCB. We will then enable the user to make more elaborate markups such as text in the virtual space. This could be done with the addition of a virtual keyboard, or possibly voice dictation if we have time to implement it. As there are more and more annotations added to the visuals, we will have to implement some sort of 3D visual management techniques to create clean layouts. We could follow something similar to the hedgehog labeling [4] technique to keep the visuals clean and organized. Finally, we will want to add a time limit on how long a mark will remain on screen, or some sort of deletion process to remove the markups.

5 Data Transmission

5.1 Networking

This project requires some data transmission to be able to collaborate with multiple headsets. We will make attempts to achieve some sort of networking, however these will all be stretch goals for the project. That being said, the first step towards this goal is testing networking through a single PC. Getting our headsets to view the same thing is a good starting point that gives a clear sign of functionality. From there, getting audio to send is the next step. This sets up basic communications between users. These headsets should view the same thing, with a single user broadcasting and the other viewing. A stretch goal should be moving the headsets to completely different networks, with a cloud-hosted server. Refining the code and getting a low latency is another stretch goal. The last stretch goal is getting the project working in Netcode, the new Unity networking standard. This would meet and exceed our clients requirements for network functionality.

5.2 Security

Since we are building a PCB collaboration program, there could be some trade secrets transferred across the network during the collaboration process. This illuminates the need for network security. It's necessary to make sure that the data won't be stolen or lost. Our client, Kevin McGrath, is an expert in network safety and he would like our project to have a secure network when transferring data. One of the steps that we will implement is encrypting the transferred data and decrypting the data at the other user's end. We would be able to secure the data from the ZED Mini by encrypting the hex data with a hash table. Some sort of AES (Advanced Encryption System) would allow us to confidently send data from user to user without having to worry about it been viewed by outsiders. Another security feature that will be implemented into our program is a user login system that only allows registered users to join the collaboration suite. This registration will contain hashing and salting of the users passwords to prevent the files from being read.

6 Gantt Chart

images/GanttChart.png

7 Bibliography

[1] "Hardware."HTC VIVE, https://www.vive.com/us/product/vive-virtual-reality-system/.

[2] "Bring Your Imagination to Life." Stereolabs, https://www.stereolabs.com/zed-mini/.

[3]"Comparison of Audio Network Protocols." Wikipedia, Wikimedia Foundation, 3 Oct. 2019, en.wikipedia.org/wiki/Comparison_of_audio_network_protocols.

[4] Tatzgern, M., Kalkofen, D., Grasset, R., Schmalstieg, D. (2014). Hedgehog labeling: View management techniques for external labels in 3D space. 2014 IEEE Virtual Reality (VR), 27-32.

geometry textheight=9.5in, textwidth=7in

[parfill]parskip

CS Capstone Problem Statement

May 16, 2020

Augmented Reality Collaboration

Prepared for

Oregon State University Instructor

Kevin D McGrath

Signature Date

Prepared by

Group CS 13

ARC

Carson Pemble

Signature Date

Ryan Miura

Signature Date

Haozhe Li

Signature Date

Abstract

Our client faces the problem of collaboratively annotating Printed Circuit Boards (PCBs) from remote locations. There is currently no technology to facilitate this at a reasonable price, so we propose using a Zed Mini camera attached to an HTC Vive along with the software we will be developing will allow for our client to annotate PCB boards in VR (Virtual Reality). Our software will allow users to collaborate on board designs and view board physicality independent of one another, while sharing visuals, audio, and user added virtual markups. We will measure our progress through iterative prototypes, allowing our client to test our work. This will result in saving the Oregon State University research department from unnecessary expenses, furthering the advancement of affordable AR (Augmented Reality) technology, and help our client continue his research.

CONTENTS

8 Problem Description

The problem we are dealing with is that there's no easy way for colleagues to collaboratively annotate a PCB board remotely. A PCB board is used to support electrical components and to connect them together. These components are usually soldered onto the board to fasten them [1]. Annotating these boards involves editing the schematic information as the physical layout is iterated upon [2]. Currently, the process for completing this work requires all parties to be present at the same location. A physical or photographic inspection of the board must be used.

While this works, it restricts schedules and prevents efficient work from getting done. Users can send pictures, take videos, or even video call to show off a board, but this lacks a certain amount of control and efficiency that could be improved upon. Allowing each user to be able to view and interact with the board in real-time would increase the amount of collaborative work that a team could complete. Also, being able to handle a single board independent of other workers without making copies would cut back on resources used while increasing productivity.

Current technology capable of this exist, such as the Microsoft HoloLens, but they can cost around $4000. This amount is not affordable to all, which eliminates this possibility between colleagues. Without funding to purchase enough headsets, those wanting to collaborate on a board will have to view it in the ways described above. A lower cost alternative could solve all those problems.

9 Proposed Solution

The proposed solution involves VR headset and a Zed Mini camera. A Zed Mini camera will be attached to the front of the VR headset. This will allow the user to simulate a first-person interaction with a PCB board in an augmented reality for the other viewers. This is because the camera will be set at the correct interpupillary distance to mimic what the current user is seeing. This camera feed will be transmitted and used by our software to display the same PCB board in real-time to the other users who will be viewing the displayed feed on their own headsets. These other users will be required to have their own headset system with a Zed Mini attached for when they share visuals. This setup will allow multiple users to view the same PCB board from remote locations, solving the previously stated issue.

All users will be able to see and annotate the PCB board they are viewing. This will be implemented using a fiducial marker so that the PCB board can be oriented. The Zed Mini can use markers on the corners of the board to determine the shape and geometry of it. This will allow a coordinate system to be mapped to the board, which the virtual notes and mark ups will be located on.

A more complex solution and a stretch goal for our project would be to use the markers to scan the design of a PCB board and create 3D objects in VR that correlate to that PCB board. This would allow multiple users to be able to interact with the same PCB board independent of the other users. Before that, a user should be able to annotate and mark up the live stream of a PCB board from another user.

This will be done using AR annotations and the mapped coordinate system. The user will be able to choose options from a 3D object interface, which will be later developed in Unity. This will provide a convenient way to communicate about a board's design. All users will be able to use their VR controllers and add comments and make mark ups on the board in real time. These markings will also have a way to manage their lifetime on screen, to reduce annotation clutter.

The board and the markings made will be shared between multiple headsets hundreds of miles apart. They will share video and audio streams, using network protocols to broadcast to each other. The broadcasts must be fast with low latency to simulate real-time interactions as much as possible. The broadcasts will also be secure, allowing for safe data transfer. At the end, these headsets will be able to perform all these actions quickly between colleagues in Oregon and Texas.

10 Performance Evaluation Metrics

The best way to measure progress in this project would be through iterative working prototypes. Since our client has never used technology like this, it would be difficult to use a performance percentage as no previous data exists. Delivering something our client can pick up and use, despite its faults or bugs, would give them the best understanding of our progress and how far we are from a finished product. This could be done through video conference calls, or in person meetings with our client when he is in Corvallis.

Using iterative working prototypes allows us to define some steps, goals, and stretch goals for the project. Getting a single headset up and running with video transmission would be a smart and simple first step to take. From there, connecting multiple headsets together through a network and streaming the video feed and audio to all of them would be a good next step. Then, making a switch from streaming video to scanning the PCB board in VR would be the next prototype to come out. This prototype is the most complex and thus the most bug-prone of them all, so it would take some time to get this working. But after that, the next step would be to get the multiple headsets working and able to receive the same board model from the original with annotation functionality. Finally, a stretch goal would be creating a way for use to scan the board and grab the schematics and other board details from the internet and projecting that information on the headset screen. These prototype goal points will outline our work over the upcoming months of this project.

While this provides a good sense of progress to our client, a way we could improve on existing products would be cost. Microsoft has developed a similar project, the HoloLens, around the price of $3500. One method of evaluation we could focus on is cost. With the ZED Mini and the HTC Vive being priced at $500 each, we should be able to make at least three headsets for the same price as one HoloLens. If we are able to achieve this goal of making three functional headsets, the ARC team and our client would be very pleased.

11 Bibliography

[1] En.wikipedia.org. (2019). Printed Circuit Board. [online] Available at: https://en.wikipedia.org/wiki/PrintedCircuitBoard Accessed 12 Oct. 2019

[2] Resources.orcad.com. (2019). What is Annotation and Back Annotation in PCB Design?. [online] Available at: https://resources.orcad.com/orcad-blog/what-is-annotation-and-back-annotation-in-pcb-design [Accessed 12 Oct. 2019].

geometry textheight=9.5in, textwidth=7in

[parfill]parskip

CS Capstone Design Document

May 16, 2020

Augmented Reality Collaboration

Prepared for

Oregon State University Instructor

Kevin D McGrath ______________________ ________
Signature Date

Prepared by

Group CS 13
ARC

Carson Pemble ______________________ ________
Signature Date

Ryan Miura ______________________ ________
Signature Date

Haozhe Li ______________________ ________
Signature Date

Abstract

To complete our project and accomplish our project's goals, we have divided up the work among our team members and have agreed upon how we will develop the augmented reality collaboration application for our client. After thorough research, we have documented the best known method for completing our project withing the current time frame. This document serves as a detailed guide on how to implement this project and what the rest of this year is going to look like for our team.

CONTENTS

12 Introduction

The purpose of this document is to provide a detailed guide on how to implement the project. The document will identify specific components of the project, explaining the steps taken to implement each.

This document is limited to the nine key components discussed in the tech review document. These were determined to be the most vital elements of the project. All nine components will be discussed in detail and explain how the implementation of each are vital to our project.

13 Hardware Design

The first steps in developing our project includes the creation of our specific Augmented Reality (AR) headset. This includes the combination of a 3D Stereo camera and a Virtual Reality (VR) Head Mounted Display (HMD).

13.1 AR 3D Camera

In this project, we will be using a Zed Mini camera that can capture 3D stereo images and video. Our client has worked with this camera before and thus felt it was a good choice for this project. The Zed Mini will be able to stream video to other users at a base level, but includes many other useful features that will be helpful. These will be accessed through the Zed SDK.

To capture video from the Zed Mini, a Camera object is created using the Zed SDK. The Camera object is configured with an InitParameters object, allowing for a large number of configurable parameters [14]. The camera is then opened with the open function [15]. To stream video, a StreamingParameters object must be created, allowing for configuration of properties such as the port and bitrate. Then, the enableStreaming function is called to enable the Zed streaming module. With streaming enabled, the grab function used to capture video will also send the frame over the network.

To receive video from another Zed Mini stream, an InitParameters object must be initialized with the IP and port of the sender. The camera is then opened with the open function and the InitParameters, allowing the application to function as if the sender's Zed Mini were attached to it. The grab function can be used to grab the current frame and the application can send it to the HTC Vive for display.

The Zed Mini also includes depth sensing, which gives a relative Z value to each object captured by the camera. This information can be passed from the API to our program. The depth mode and units can be configured in the InitParameters object. When the current frame is grabbed, the retrieveImage function is called to grab the image seen in the left camera eye. Then, the retrieveMeasure function is called to grab the depth map generated by the camera. The retrieveMeasure function can also be used to get a 3D point cloud or surface normals. The surface normals can be used by the program to determine the orientation of the PCB board at any given time. Other features included with the Zed SDK are positional tracking and spatial mapping, but these won't be used in this project.

Testing of the cameras will start with simple functionality tests. The cameras will be given to us by the client, as he already has a number of them. Then, the video stream could be created locally and looped back to the headset, allowing the user to see the world around them. This would be a successful test of the streaming functionality. Receiving the video would require multiple headsets attached to the same machine, then moving them to separate machines and networks. If problems are encountered with the camera itself, the Zed SDK includes a diagnostics tool that checks the camera and reports any problems with it. The Zed Mini support department will have to be contacted if the SDK cannot be repaired by the SDK.

13.2 VR Headset

13.2.1 Description

The VR headset comes as a complete unit, and thus does not require much assembly. However, this project requires a few modifications to the base unit, so those will be detailed here. This part of the project will be done at the end of Fall term, beginning the week of November 25th and ending three weeks later. This is due to the importance of the headset in the rest of development. While unit tests can be run on code and visualizations can be seen on a monitor, testing on the headset is the easiest way of indicating whether the software works.

13.2.2 Hardware

First, the headset itself must be acquired and examined. The client will work alongside the team to secure an HTC Vive. The headset itself will only require a test run to make sure that it works. First we will find a room large enough to set up the base stations, and with a computer that is powerful enough to run our VR headsets. Then we will set up the link box to our computer by connecting an HDMI cable to the link box and the PC's graphics card, and by connecting a USB cable from the link box to the a port on the PC. Then the we can connect the headset's 3-in-1 cable into the link box with the HDMI, USB, and power ports. At this point the VR headset is set up and ready to be tested.

The next step for developing our project is to connect our 3D Stereo camera to our VR headset. A ZED Mini camera must also be secured by the team and our client. The headset will be modified by mounting the ZED Mini on the front of it. The ZED Mini comes with an attachable mount compatible with the HTC Vive. This mount will be placed onto the HTC Vive followed by the ZED Mini [12]. A USB-C cable will connect the ZED Mini to the headset, providing power. This is the only physical addition we will have to make to our headset. After these modifications, the headset is ready to be used for testing of the software.

13.2.3 Software

Developing software for the HTC Vive will be done in Unity. An SDK that will make VR/AR development easier is the SteamVR SDK, which comes with many useful features.

One of the features it gives is different views. The view can be changed within Unity to view how different physical components are monitored by the system. By putting the headset on, the headset and controller movement can be viewed through small markers on screen. This will be useful in finding and using motion to interact with the app. Within Unity, actions can be defined based on controller input via buttons or motion. These actions get stored in a JSON file, but the GUI provides a user-friendly way of editing them. Individual actions are put into action sets and bound to a controller. Again, Unity provides a GUI for this, giving a list of possible points of interaction for each button. These actions will be used to map buttons to functionality within the program, such as making, editing, and deleting annotations.

While this meets most of the requirements we need for the project, another useful feature that can be incorporated is a laser pointer. Using different colors for different users will allow our clients to indicate a specific component to discuss without adding an annotation. A cube object is created and stripped of most of it's original functionality. A C script is used for the action to show or hide the laser. It also handles displaying the laser and scaling it so it looks correct to the human eye. The laser is then set to one of the controllers so it appears as though it's coming from the user's hand. This gives the user a good indication of what they're pointing at. For other users' lasers, a hovering point on the board would clean up the clutter of multiple lasers on screen. This can be achieved by using the ZED Mini's SDK to grab depth data. Then, the point can be drawn at the point the laser crosses the board's depth.

13.2.4 Testing

To test the connection itself, the ZED SDK will be used. Once the SDK is downloaded, the ZED Explorer program will be used. This will allow the ZED Mini to stream it's video feed to the HTC Vive headset. Once a video feed can be seen, the functionality of the headset can be confirmed. There will be more on this process later on in the document.

13.3 AR Headset Audio

Our software will have multi-user voice communication capabilities, and thus will need to take audio hardware into consideration. The HTC Vive includes a 3.5mm stereo audio headphone jack, so any standard headset will work with it. In addition, the Vive includes headphones with each unit, so that will eliminate the need to purchase audio hardware. There is an integrated microphone for audio input on the headset that can be used to capture voice communications, thus external microphones will not have to be used. This design will allow users to choose their own audio hardware without having to worry about a microphone.

In order to ensure that the audio equipment works properly, a software called SteamVR is perfectly compatible with the Vive VR headset. SteamVR can troubleshoot the hardware device and if the software detects any problems, steamVR will provide an error code explaining what is wrong. There is a lot of documentation and troubleshooting information for the HTC Vive but if the user still can not find a solution to the particular problem; it may require the user to send a system report to the SteamVR mailing list so Valve can diagnose and hopefully solve whatever is plaguing the system or device setup[13]. It means that if the user encounter other difficulties, they can submit the software generated report to the manual support center by email.

14 SOFTWARE DESIGN

The second and more challenging piece of developing our project is the creation of our software application. A majority of our project work will be focused on this section. This will include the setting up of our development environment, creating the basics of our application, improving the user experiance, securing the transferred data, and configuring the network connection.

14.1 Visual Software and AR Software Development Kit

After researching many options for our 3D visual software development, we have decided to use Unity 3D for many reasons. The software is free, easy to learn, and the ZED Mini has a plug in that works well with Unity. The first step to creating our AR collaboration application will be to download Unity 3D onto our developer and testing machines. If we use the computers in Batchellor Hall at Oregon State University they have powerful graphic cards and the latest version of Unity 3D already installed. Therefore, all we would have to do is download the ZED Plugin for Unity which will allow us to actually use the stereo pass through from the ZED Mini and start development of our AR application. This plug in can be installed as a Unity package and immediately we will have access to assets, scripts, and sample scenes [1]. Once the Unity basics are set up the next step would be the AR Software Development Kit.

Not all visual software programs can work in 3D and not all 3D programs allow for development in AR. Research shows that Stereolabs has a SDK specifically developed for the ZED camera's allowing us to add depth and motion sensing to our application and many other features all for free. The software is available from the ZED website as well as the documentation for this kit. We will download the latest version of the ZED SDK onto our developer personal computers (PC) and import the ZEDCamera.package into Unity [1]. We can then read through the documentation to get caught up to speed on how the ZED Mini is developed and how we can use that to benefit our project.

Once the SDK is installed we can open and run the ZED Explorer. This will give us high definition (HD) video at 720p with 60 frames per second (fps) [1]. This will also automatically recognize the HTC Vive controllers and allow us to use them flawlessly. After all of that is set up and complete, we will be able to launch ZED World. This is the piece of the software that will allow us test and experience applications in the mixed reality world. We will start actual coding development on our project by creating a new 3D Unity project and making sure all of the ZED packages and plugins are downloaded and installed. Then we can create a new scene and add the AR camera prefab to the scene and enable positional tracking (which can be done in the inspector panel). These are key to starting our AR application.

A large requirement that our client has requested from us is that a user can place an object, like a circle, on the board to identify a pin or chip. To do this we need to use plane detection and object placement techniques. We will use the ZED mini to perform a scan of the real-world environment and create a mesh plane on the PCB. This will be done by creating a plane detection manager object that will create other GameObjects based off of different surfaces and add rigid bodies to these newly created objects. Then we will create arrows, circles, and other annotation objects that the user can place with the click of a button. All other annotations will be a variation of this.

14.2 Audio Software

To broadcast audio through the program, an audio broadcasting software must be used. WASAPI will be used for this project, as it's supported by Windows which is our developer OS (Operating System) of choice. A separate class for handling audio broadcasting will be used. All functionality will be abstracted to this class, allowing for all audio concerns to be handled by this one class for ease of code management.

The process of sending an audio message will begin by creating a WASAPI capture client object, activating it, and initializing it. This initializes an audio stream to be used to collect an audio sample. The audio stream will be connected between the software and the microphone on the headset. The capture client will get the size of the next buffer object using a built-in method. The buffer will be stored in a variable of the correct size. Then, the buffer will be released. Each buffer variable will be sent to the other users through a UDP package transfer. This process will continue until there is no more audio to be sent. However, since this will be an audio stream, this process will continue throughout the duration of the call.

The process of receiving an audio message will begin by receiving the sent UDP packages. These packages will contain the audio information to be played from the headphones. A WASAPI render client object will be created, activated, and initialized. Then, a built-in method will be used to get the default playback device of the computer. This will be the headphones used with the headset. The client will then use another built-in method to get a pointer to the next open buffer space in the returned playback device. The buffer will be filled with the next UDP package, and then the buffer will be released to play the audio back. This process will continue until there is no more audio to receive. However, since this will be an audio stream, this process will also continue throughout the duration of the call.

Both processes will be continuously running on all devices. The UDP package sending will have to send to all users, while the receiving part will have to receive from all users. This will be handled using a users array that keeps track of the current members of a session. This allows for the number of members to increase or decrease without affecting package distribution. Another problem can arise from receiving and playing audio from multiple users at the same time. This will be mitigated by adding packages to the buffer by arrival time. Each message sent will already be received in terms of arrival time, so this will not take much effort. Also, since the application is not focused on audio streaming, some distortion will not be cause for concern.

Testing this will require two or three endpoints to be set up. Testing with two connections at first will result in simpler tests overall. These tests will simply record for 5 seconds, capture the data, and send it to the receiver. This will be a simple way to test the entire audio capture class outside of unit tests. Then, scaling these tests out to more than two users communicating in real-time would simulate the conditions for a completed project. The UDP package transfer will also have to be tested on its own. This will mainly consist of ensuring the data is not changed during the transfer process.

14.3 Securing the Data

Security will be implemented using the AES (Advanced Encryption Standard) class by Microsoft. This is a symmetric cipher with a single password/key. This class works by having a function that will encrypt the data and a function that will decrypt the data. Encryption is done by converting the data into Hexadecimal format, divided into blocks of data, combine the blocks with the key, and then passed through the AES algorithm to create ciphered blocks. We will be using a 128-bit key to balance security with performance speed of our software.

This software package has code written for encryption and decryption in the AES protocol, making it easier to implement and use. Within our project, there will not be a separate class will contain the security class, but it will be injected into all other classes it's needed in using dependency injection.

The first part of using AES is getting the key. This key will have to be known by all users of the software. If a user does not have access to the key, the software will exit. This key can be kept in a user secrets file, with each user owning a copy of the file. This seems to be the easiest way to manage this key. It also allows our client to distribute the key to other users as they see fit, allowing for scalability in the future.

The encryption process will occur every time before the data is sent between two users. Obviously the data must be encrypted before it is sent to another user to keep it secure. The data to encrypt must be in string form, but once converted it will be passed to an encryption function where the data will be encrypted and returned to the user. This is done using the encryption key and the built-in Microsoft AES class. The data will then be sent to any other users in the session.

The decryption process will occur every time data is received from another user. The data must be decrypted before it is used in the application. The data to decrypt must also be in string form for it to be passed to a decryption function where the data will be decrypted and returned to the user who received the encrypted data. The process is the reverse of the encryption process. After decryption, the data can then be used in the application.

Testing security will include unit tests and visual inspections of files being transferred. A simple visual test of security would be simply writing the package out to the console before and after encryption. If the contents before encryption match the contents after decryption, this shows the encryption and decryption process is working as intended. Also, visually inspecting the contents before and after encryption allows for verification of encryption. Passing in a readable string to the encryption function would be an easy test, since verification would only involve printing the encrypted string to the console and visually inspecting it. If the encrypted string is impossible to decipher quickly, the test will pass. Much like the audio software, since security is not a main concern in this project, the extensiveness of tests will be limited to ensuring the AES security class is functioning as intended. Trying to hack the software for testing will be outside the scope of this project.

14.4 Creating a Network

The transmission of 3d media streams will involve knowledge of the network. How to establish such a network connection is a problem. Our research will start with the data type API of 3d media stream. Study whether there are network interaction methods and software that can be used.

In order to exchange information, we need to find the right way to connect to the network. First, we need to figure out how to transfer the information among the internet. Data travels across the internet in packets. Each packet can carry a maximum of 1,500 bytes. Around these packets is a wrapper with a header and a footer. The information contained in the wrapper tells computers what kind of data is in the packet, how it fits together with other data, where the data came from and the data's final destination.[7] That means we can insert the information in a packets with different protocols and then edit the format of the information. The format always depends on the data type and API (a Application Programming Interface). Before we actually establish a network connection, we need to choose the right network software to detect the network protocol and the transmitted data.

Some of the more popular examples of networking software include Logic Monitor, Datadog, Vallum Halo Manager and ConnectWise, among others. As always, it is recommended that the user takes the time to explore all the various options available and consult customer reviews before deciding on a particular product.[8] Most of these software either ask for some personal registration information or is not free. In another article called "10 Best Network Monitoring Tools & Software of 2019", it suggests some better network software.[9] We choose the software called: PRTG Network Monitor from Paessler. With the network monitor software, we will know if we send or receive the correct information by our software in the future.

Finally, based on different data transmission types, the corresponding network protocol should be selected. The data transmitted by this software should be a video audio stream. So we did research on protocols for media streaming. Unlike TCP, will not spend extra efforts on fixing errors with delivery, it'll proceed with sustaining uninterrupted flow of information. This feature makes UDP more suitable for live video streaming. However, due to the fact that TCP is widely used for various activities on the web, UDP transport protocol might be blocked by some firewalls.

Furthermore, TCP is a preferable option for streaming video on demand or for those broadcasts when small delays do not make a big difference. [10] Therefore, my conclusion is that in the future we should use the UCP network protocol that does not accumulate transport packets and use PRTG Network Monitor software to detect and improve network connectivity.

14.5 Virtual PCB Information

One of our stretch goals is for the user to be able to import a specific board schematic file and then have our software convert that into usable data for the user to identify the pieces and see information by just selecting the chip. If we want our application to be able to identify the key chips on the board, we will have to first use CAD (Computer-Aided Design) software to backtrack the files and related them to the boards we will be observing. After researching it appears that the best technology for this project would be to use DipTrace. This software has very detailed and complete 3D previewing would be extremely helpful for attempting this extra goal.

We can buy and download DipTrace from online from the DipTrace website. The key feature of DipTrace that we would be using is the 3D modeling and real-time preview features. After downloading and installing DipTrace we will then have to download the 3D libraries to allow us to use these features. These libraries are also available from the DipTrace website and we will be able to obtain 3D patterns, components, and all the items required to create a 3D rendering of the user imported schematic file.

We will do this by first having the user import a schematic file to our software. We will then have our software take that ASCII file, or P-CAD file, and import it into DipTrace [3]. Then we just open the tools tab and launch the 3D preview viewer. Once DipTrace creates the 3D model we will then exported and imported into Unity 3D as a new object. This way the user can rotate the board, zoom in on certain sections and see more detailed chips on the board.

15 Project Timeline

images/GanttChart.png

16 Conclusion

This document includes our plans for the development of our project. Things may change as the year goes on and more time is spent in certain areas, while other steps may take less time than what is currently planned. Overall, this is our road map for the setup of our hardware and the setup for our software environment. We will create the desired product for our client by following the steps previously stated within this document and have it completed by the dates above.

17 Bibliography

[1] Stereolabs.com. (2018). Unity - Getting Started with Unity — Stereolabs. [online] Available at: https://www.stereolabs.com/docs/unity/ [Accessed 30 Oct. 2019].

[2] Stereolabs. (2018). Building Multiplayer AR Experiences with ZED Mini. [online] Available at: https://www.stereolabs.com/blog/stereolabs-brings-multiplayer-ar-to-vr-headsets/ [Accessed 8 Nov. 2019].

[3] Diptrace.com. (2019). 3D Modeling - DipTrace. [online] Available at: https://diptrace.com/diptrace-software/3d-modeling/ [Accessed 8 Nov. 2019].

[4]. "Stereo Camera." Wikipedia, Wikimedia Foundation, 25 Sept. 2019, en.wikipedia.org/wiki/Stereo_camera.

[5]. Getting Started - Introduction. (n.d.). Retrieved from https://www.stereolabs.com/docs/getting-started/.

[6]. "Oculus Rift vs. HTC Vive." Oculus Rift vs HTC Vive — Which VR Set Is Best for You? - Newegg.com, https://www.newegg.com/vr/guides/oculus-rift-vs-htc-vive.html

[7]. Stricklan, "How IP Convergence Works." HowStuffWorks, 8 Mar. 2010, https://computer.howstuffworks.com/ip-convergence2.htm.

[8]. Beal, Vangie. "Network Software." What Is Network Software ? Webopedia Definition, https://www.webopedia.com/TERM/N/network_software.html.

[9]. Wilson, Marc, and Marc Wilson. "10 Best Network Monitoring Tools & Software of 2019: FREE - UPDATED!" PC Network Downloads - PCWDLD.com, 30 Mar. 2019, https://www.pcwdld.com/best-network-monitoring-tools-and-software.

[10] "How to Sort Through the Variety of Streaming Protocols." Letzgro, 25 Aug. 2016, https://letzgro.net/blog/the-variety-of-streaming-protocols/.

[11] "What Do I Do If My ZED/ZED Mini Is Not Working?" Stereolabs, https://support.stereolabs.com/hc/en-us/articles/360010101213-What-do-I-do-if-my-ZED-ZED-Mini-is-not-working-.

[12] https://www.stereolabs.com/zed-mini/setup/vive/

[13]Hesse, B. (2018, July 9). The Most Common HTC Vive Problems, and How to Fix Them. Retrieved from https://www.digitaltrends.com/computing/common-htc-vive-problems-and-how-to-fix-them/.

[14] "InitParameters Class Reference: ZED API," InitParameters Class Reference — ZED API. [Online]. Available: https://www.stereolabs.com/docs/api/structsl_1_1InitParameters.html. [Accessed: 27-Nov-2019].

[15] "Getting Started - Introduction," Stereolabs. [Online]. Available: https://www.stereolabs.com/docs/getting-started/. [Accessed: 27-Nov-2019].

textheight=9.5in, textwidth=7in

[parfill]parskip

CS Capstone Progress Report

May 16, 2020

Augmented Reality Collaboration

Prepared for

Oregon State University Instructor

Kevin D McGrath

Signature Date

Prepared by

Group CS 13

ARC

Carson Pemble

Signature Date

Ryan Miura

Signature Date

Haozhe Li

Signature Date

Abstract

Throughout this term our team has learned a lot about our project and what our goals are for this term. The ARC team has communicated with our client and have agreed upon requirements that will have to be met by the end of this year. We have encountered many issues from failure in communication to paper revisions. We have also overcome these issues with clearer communication and more team meetings. Looking back at this term we have found that we still need to improve our communication with our client and have consistent meetings with our TA and our client.

CONTENTS

18 Our Project

Out project is to create Augmented reality (AR) Printed Circuit Board (PCB) collaboration software product. With a 3D stereo camera attached to the front of a VR headset, we will create our own AR headset. Our project will allow the user to simulate a first-person interaction with a PCB board in augmented reality for other viewers to see and also interact with. This is because the camera will be set at the correct interpupillary distance to mimic what the current user is seeing. This camera feed will be transmitted and used by our software to display the same PCB board in real-time to the other users who will be viewing the displayed feed on their own headsets.

19 Project Purposes and Goals

The purpose of this software is to allow users to improve printed circuit board design through a virtual reality collaboration suite. Our goal is to be able to create this software to allow for our client and his colleagues to present information about various parts of a printed circuit board in a 3D virtual reality video conversation. With the addition of augmented reality annotations this new form of communication should allow for further advancements in our clients research and improve long distance project collaboration.

Allowing each user to be able to view and interact with the board in real-time would increase the amount of collaborative work that a team could complete for many reasons. Also, being able to handle a single board independent of other workers without making copies would cut back on resources and research expenses, while still increasing productivity.

20 Current State

The current state of our project is still in the early stages. We have studied the required technology and feel comfortable with the knowledge of how we plan on moving forward. We have recorded our research in many documents with clear instructions for the road map of this year. We have yet to obtained the hardware from the client, and therefore we have not yet started to implement the specified steps, but will be ready to start development at any time.

21 Project Problems

One of the problems that we faced with our project throughout the term was learning how to use LaTex. Before fall term non of our team members had used LaTex before and there seemed to be a learning curve that we were unaware of. It took some time to write our papers in the correct IEEE format.

Another problem that we ran into this term was proper communication. Both our team and our client had some difficulties with communication. Our client was very busy with young children and working remotely. This made it challenging to contact and meet with our client with times that could work for all of us. We the students also had a hard time finding a good time to meet because of class schedules. Poor communication caused us to miss meetings, where some thought the meeting was canceled and others did not.

The final problem we ran into this term was balancing this class with our other classes. This is our senior year and other classes demand more time than the previous years. One issue that we ran into was the class of one of our team members altered a meeting time and this created a time conflict with our previous client meeting time.

22 PROJECT SOLUTIONS

We were able to overcome the LaTex learning curve by finding some useful video on YouTube teaching us how to use the Latex format. Our TA also gave us instructions of specific key words in Latex editor that we are looking for.

The way that we are currently overcoming the communication problem is to use slack for our team communication and emails for client communication. As a team we are informing each other of our progress and then relaying that to our client via emails. When emailing the client all team members are to be CC'd to stay up to date on the information. With meeting times we figured it would be best to set time aside every week to meet with our client, our TA, and as a team. Once we set those times we must stick to them. We emailed our client with all the times that our team is available to meet and then chooses the best time slot for him or asks for an adjustment.

Finally, our team decided the best thing to do was to prioritize this class over our others because this grade effects the whole team. If a team member is not able to make it to client or TA meeting the team will fill in said person with the information they missed.

23 INTERESTING PIECES OF WORK

Something that all team members are very interested in and excited to work with is Unity 3D. We have a little experience with developing in Unity and all of us are excited to learn more about it and gain knowledge will working with this software. We are also very interested in the ZED Mini and the SDK that is on their website for us to use. There are a lot of functionalities that would be really fun to use and possibly add into our project if we have time.

24 LOOKING BACK

Week	Positives	Deltas	Actions
1	N/A	N/A	N/A
2	Chose a fun project	Submit Project Bids	Choose our project
3	First team meeting	Write Problem Statement	Talk with our client to understand the project
4	Agreed upon team standards	Team mate left and failed meetings	Try Slack for communication
5	More insight from TA	Failed LaTex formatting	Research LaTex and understand how it works
6	Early start on Tech Review	Failed client meeting	Improve communication between the team
7	Met with our client again	Need to improve our writing	Take more time to improve our first drafts
8	Improved our writing method	Failed client meeting	Improve communication with client
9	Team revision session	No communication with our client	Speak with professors about communication
10	Revised all papers	Haven't touched hardware	Get the technology from our client

Category	Description	Reviewers Comment	Action taken by reviewed group
Build	Could you clone from Git and build using the README file?	Cloning was fine but could not build due to hardware limitations. README is clear and understandable.	None.
Legibility	Was the flow sane and were variable names and methods easy to follow? Does the code adhere to general guidelines and code style?	Flow was sane and understandable. Some files were missing comments.	Comments were added to files missing them.
Impleme ntation	is it shorter/easier/faster/cleaner/ safer to write functionally equivalent code? Do you see useful abstractions?	Using relative file locations or dynamically determining file location would be better for login. Also, moving ControllerInput class functionality into other functions.	User login info stored in a relative file, making it safer and less error prone. ControllerInput class removed.
Maintain ability	Are there unit tests? Should there be? Are the test covering interesting cases? Are they readable?	Did not see any unit tests.	Decided that, due to the nature of the project, viewing the scenes through the headset would prove more useful than unit testing.
Require ments	Does the code fulfill the requirements?	The requirements established in the requirements document are met.	None.
Other	Are there other things that stand out that can be improved?	Using hash functions like BCRYPT for login.	Stuck with SHA-256 as it seems to meet the needs for our project.