

6-14-2018

PresentVR: Defeating Public Speaking Anxiety with Virtual Reality

Gurneev Sareen

Santa Clara University, gsareen@scu.edu

Hamilton Coke

Santa Clara University, hcoke@scu.edu

Reece Jackson

Santa Clara University, rjackson@scu.edu

Follow this and additional works at: https://scholarcommons.scu.edu/cseng_senior



Part of the [Computer Engineering Commons](#)

Recommended Citation

Sareen, Gurneev; Coke, Hamilton; and Jackson, Reece, "PresentVR: Defeating Public Speaking Anxiety with Virtual Reality" (2018). *Computer Engineering Senior Theses*. 117.

https://scholarcommons.scu.edu/cseng_senior/117

This Thesis is brought to you for free and open access by the Engineering Senior Theses at Scholar Commons. It has been accepted for inclusion in Computer Engineering Senior Theses by an authorized administrator of Scholar Commons. For more information, please contact rsroggin@scu.edu.

Santa Clara University
DEPARTMENT of COMPUTER ENGINEERING

Date: June 13, 2018

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY
SUPERVISION BY

Reece Jackson, Gurneev Sareen, Hamilton Coke

ENTITLED

PresentVR: Defeating Public Speaking Anxiety with Virtual Reality

BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF

BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING



THESIS ADVISOR



DEPARTMENT CHAIR

SANTA CLARA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING
SENIOR DESIGN

Present VR: Defeating Public Speaking Anxiety with Virtual Reality

by

Gurneev Sareen
Hamilton Coke
Reece Jackson

Santa Clara, California
June 14, 2018

PresentVR: Defeating Public Speaking Anxiety with Virtual Reality

by

Reece Jackson, Gurneev Sareen, and Hamilton Coke

SENIOR DESIGN PROJECT REPORT

Submitted in partial fulfillment of the requirements
for the degree of
Bachelor of Science in Computer Science and Engineering
School of Engineering
Santa Clara University

Santa Clara, California

June 13, 2018

PresentVR: Defeating Public Speaking Anxiety with Virtual Reality

Gurneev Sareen

Hamilton Coke

Reece Jackson

Department of Computer Engineering

Santa Clara University

June 14, 2018

ABSTRACT

In today's society, the vast majority of individuals are afraid to speak in front of a public crowd. This could range from a classroom presentation to a thousand person amphitheater, and it is because of this fear that people are more anxious in public settings where they present their own idea. In the past, people would try to tackle this problem by practicing in front of a mirror, or having friends help practice. However, there was no convenient way of practicing in front of a live audience, and we believe that with the advances in technology, especially virtual reality, that we can help solve this problem. Our idea, PresentVR, aims to provide the user with an application that places them in front of a realistic audience in virtual reality equipped with the aids and tools they would use in real life. Our solution composes of a virtual reality application where the user gives the presentations, a web application where the user can upload presentation materials, and a mobile application for additional users to join the audience and help create a more realistic environment. Succeeding in this overall system will provide people around the world with the ability to practice presentations and improve their public speaking skills from the convenience of their home.

Acknowledgements

We would like to thank our advisor Professor Amer for all his guidance, time, and patience throughout our senior design project. We would also like to thank Professor Max Sims for his expertise in all things virtual reality and his kindness in letting us use the equipment in the Imaginarium. Lastly, we would like to thank the Santa Clara engineering school for fostering a supportive learning environment.

Table of Contents

1	Introduction	1
1.1	The Problem	1
1.2	Other Solutions	1
1.3	Our Solution	1
2	Overall System	2
2.1	Requirements	2
2.1.1	Functional	2
2.1.2	Non-Functional	2
2.1.3	Design Constraints	2
2.2	Use Cases	3
2.3	Activity Diagram	6
2.4	Architecture Diagram	6
2.5	Technologies Used	8
2.6	Design Rationale	9
3	System Design for Virtual Reality Application	11
3.1	Overview	11
3.1.1	Activity Diagram	11
3.2	Implementation	12
3.2.1	Asset Creation	12
3.2.2	Main Menu	13
3.2.3	Venue Select	13
3.2.4	Presentation	14
3.2.5	Character Animation Model	16
3.2.6	Presentation Aids	18
3.2.7	Server and Communication	18
4	System Design for Web Application	20
4.1	Overview	20
4.1.1	Activity Diagram	20
4.2	Implementation	21
4.2.1	User Interface	21
4.2.2	Application State Model	21
4.2.3	Server and Communication	22
5	System Design for Mobile Application	23
5.1	Overview	23
5.1.1	Activity Diagram	23
5.2	Implementation	24
5.2.1	User Interface	24
5.2.2	Server and Communication	24
6	Testing and Results	26
6.1	Alpha Testing	26
6.2	Beta Testing	26
7	Issues and Constraints	28
7.1	Ethical	28
7.2	Social	28
7.3	Political	28
7.4	Economic	28
7.5	Health and Safety	29

7.6	Manufacturability	29
7.7	Sustainability	29
7.8	Environmental Impact	29
7.9	Usability	30
7.10	Lifelong Learning	30
7.11	Compassion	30
7.12	Risk Analysis	30
8	Conclusion	31
8.1	Summary	31
8.2	Lessons Learned	32
8.3	Future Potential	32
	Appendices	34
A	Server REST API Reference	34
A.1	UE4 Endpoints	34
A.2	Web Platform Endpoints	34
A.3	Mobile Client Endpoints	34

List of Figures

1	A use case diagram for the entire system	3
2	Activity Diagram for Entire System	6
3	Architectural Diagram	7
4	iOS Audience Control Page	11
5	Classroom Assets modeled after Engineering 326	12
6	Conference Room Chair	13
7	Conference Room Table	13
8	Main Menu	14
9	Select Classroom	15
10	Select Conference Room	15
11	Start Presentation	16
12	Pause Presentation	16
13	Audience Animation Logic	17
14	Audience State Machine	18
15	Web Activity Diagram	20
16	Web Dashboard UI	21
17	Web Create Presentation UI	22
18	Web Login UI	22
19	Mobile Activity Diagram	23
20	iOS Enter Room Code	24
21	iOS Audience Control Page	25

List of Tables

1	Risk Analysis Table	31
2	UE4 REST Endpoints	34
3	Web REST Endpoints	34
4	Mobile REST Endpoints	34

1 Introduction

1.1 The Problem

In today's society, the vast majority of individuals are afraid to speak in front of a public crowd. This could be anything from a classroom presentation to a ten-thousand person keynote. Because of this fear, people are more anxious in public settings where they present their own ideas, such as addressing coworkers, negotiating with a boss, or any other situation where they have the spotlight. Modern companies realize that creativity and innovation are paramount to their success. Therefore, the ability of individuals to confidently speak and express themselves not only benefits them, but benefits their.

1.2 Other Solutions

The current options for practicing speeches are in front of the mirror, or recruiting friends as an audience to practice. Speaking in front of a mirror can help you improve your body language, but it is a poor way to get over the anxiety that comes with speaking in front of a large group of people. However, scheduling with a large number of people is often difficult, and there is no way to have a large group and keep it realistic. People can also sign up for memberships at organizations like Toastmasters, where they can meet with other individuals to work on their public speaking skills. But this also costs money and cannot be done from the comfort of your own home. Essentially, there is no current way to practice actually speaking in front of a large number of people conveniently.

1.3 Our Solution

Our solution uses the realism of Virtual Reality to simulate speaking in front of an audience at the convenience of the user. The user can define what type of audience he/she wants to speak in front of and then be placed in a virtual simulation of said audience. Also, with the option to speak in front of different venues, such as a classroom or conference room, the user can practice different presentations for different environments. Additionally, the audience can mimic different moods and elicit specific feedback to better prepare the user for their speech. Providing this unique experience accustoms the user to speaking in front of a crowd and helps eliminate the anxiety they would have when they present their ideas in reality. Furthermore, there is no need to gather friends or awkwardly stare at yourself in the mirror when preparing for a speech. Finally, there can be a solution for individuals to practice speaking in very realistic settings and develop confidence in their ability to present ideas, thereby eliminating their anxiety with speech.

2 Overall System

2.1 Requirements

This section outlines the requirements and design constraints that our system requires to meet our goals. A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. Design constraints are the rules or limitations through which the design is created or conceived.

2.1.1 Functional

- System to have a method for the user to simulate a presentation.
- System to be able to simulate venues or presentation rooms with different sizes and atmospheres.
- System to have the ability for users to upload presentation documents.
- System to be able to simulate a realistic audience.
- System has a way for user to control where they are in their presentation in regard to notes and slide deck.
- System has ability for the audience mood to be changeable.

2.1.2 Non-Functional

- Ability for the system to be user friendly.
- Ability for the system to be easily accessed by presenter and audience members.
- Ability for the system to be easily maintainable.
- Ability for the system to be visually appealing.

2.1.3 Design Constraints

- Ability for the system to run on the Oculus Rift.
- Ability for the system to interact with a mobile application.
- Ability for the system to interact with a web application to upload files

2.2 Use Cases

The use case diagram shows what actions each actor in the system can perform. They are accompanied by a description of each action to ensure that all functions of the system are clearly defined and the design is complete.

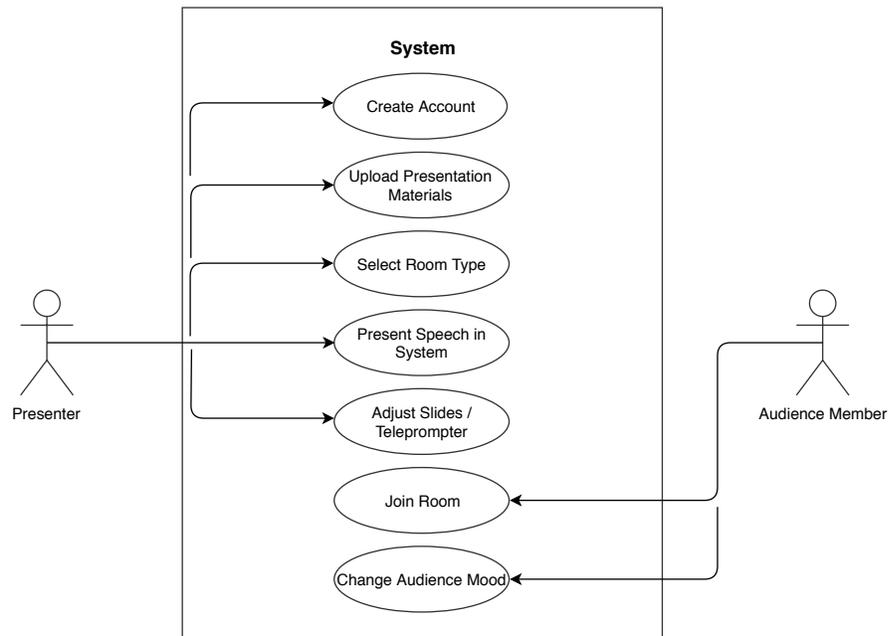


Figure 1: A use case diagram for the entire system

Overview:

1. Create Account-

Goal: For presenters to be able to save their data and documents uploaded so that they can access where they left off upon log in. Additionally help to customize the application to their profile. Audience members will be able to log in and create their own avatar if they wish; it would not be mandatory.

Actor: Presenter and Audience Member

Preconditions: None.

Steps: Presenter or Audience member to click create account button.

Postconditions: Account has been created.

2. Upload Presentation Materials-

Goal: To allow presenters to have their speech and slide deck visible while presenting to give a realistic experience.

Actor: Presenter.

Preconditions: Account created.

Steps: Presenter to click add documents button and upload a text or PowerPoint file on the web application after logging in. Must use the room code on the

Postconditions: Documents have been added to current presentation.

3. Select Room Type-

Goal: To allow the presenter to practice presentation in different environments.

Actor: Presenter

Preconditions: Account has been created.

Steps: Presenter to click new presentation button and select room type after the optional step of uploading documents.

Postconditions: Presentation has been created and presenter is now in the room, ready to practice.

4. Present Speech in the System-

Goal: To practice speech in order to get over speech anxiety in a realistic environment.

Actor: Presenter

Preconditions: New presentation has been created.

Steps: Click present on a created presentation.

Postconditions: Presenter to return to main menu after exiting the practice.

5. **Adjust Teleprompter and PowerPoint Slides-**

Goal: To allow the presenter to have presenter notes and view the slide they will be presenting on to simulate a realistic environment.

Actor:Presenter

Preconditions: Presentation has been created and presenter is presenting.

Steps: Presenter to press buttons on the hand remotes the VR system comes with.

Postconditions: Teleprompter or slide presentation has moved to next page or slide respectively.

6. **Join Room -**

Goal: To allow the audience members to join a presentation in order for the presenter to gain realistic feedback and have others view the presentation.

Actor: Audience Member

Preconditions: Presentation has been created, room has been joined, and code has been generated.

Steps: Presenter to send the 'add audience member' code to individuals who want to join in on the speech. The audience member then enters that code on the mobile device to join the room and start changing the mood of the audience.

Postconditions: Audience members have access to join the presentation.

7. **Change Audience Mood-**

Goal: To allow the audience members to change mood of the audience members in order to simulate a realistic audience.

Actor: Audience Member

Preconditions: Presentation has been created and audience member has joined the room.

Steps: Audience member to change mood on their smart phone using the mobile application.

Postconditions: Audience members have changed the audience members' mood.

2.3 Activity Diagram

Below is the activity diagram for the entire system. As seen in figure 2, there are two entry points, one for the presenter and one for the audience member. The presenter opens up the Virtual Reality application where he or she can send the room code to the audience member or upload materials to their presentation. To upload the materials, the presenter must log into the web application with the generated room code and add the necessary files there. Once uploaded, the presenter can then go back to the VR application and begin their presentation. Once the audience member has joined the room, they can change the mood of the audience by using the mobile application.

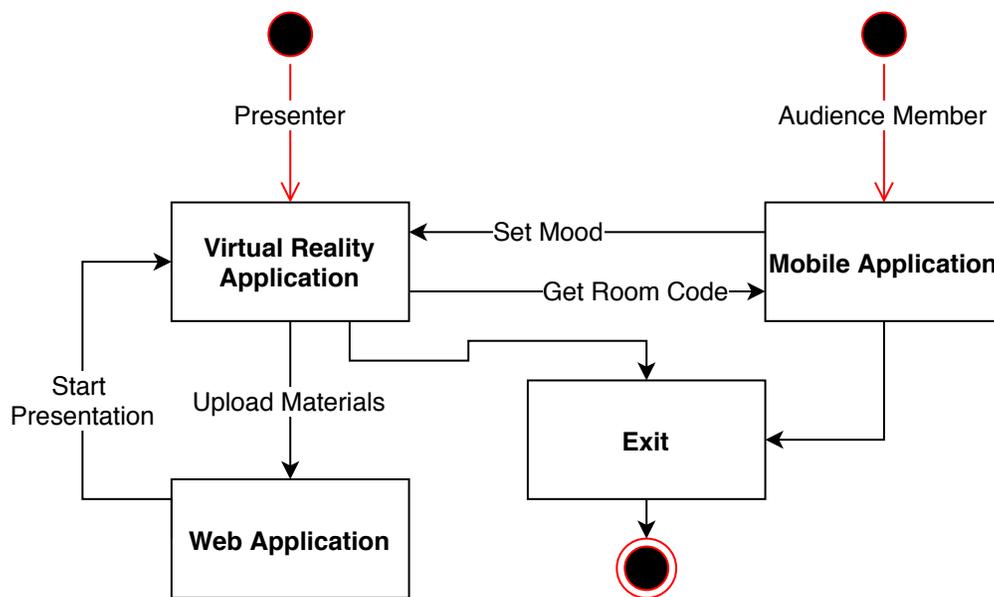


Figure 2: Activity Diagram for Entire System

2.4 Architecture Diagram

The architecture we are using is a client-server architecture, shown in figure 3. Client A is the user's computer, which will be running our system. The user will have a head-mounted display (HMD) connected to their computer, through which they interface with our application.

Audience members interact with the user's presentation through a mobile interface. In order to support this real-time remote audience, Client A will be connected to a remote server. We will have a database on the same machine as the server to store necessary information. The audience members' mobile devices communicate with the server through HTTP protocols. The server then in turn communicates with Client A to provide input into the user's application.

The presenter also has the option to upload presentation materials such as PowerPoint slides or text for note cards to help aid them during their presentation. Client C will upload the documents the presenter will use via the web application, which will then communicate with the server to send the materials to Client A, making the materials available in the virtual system.

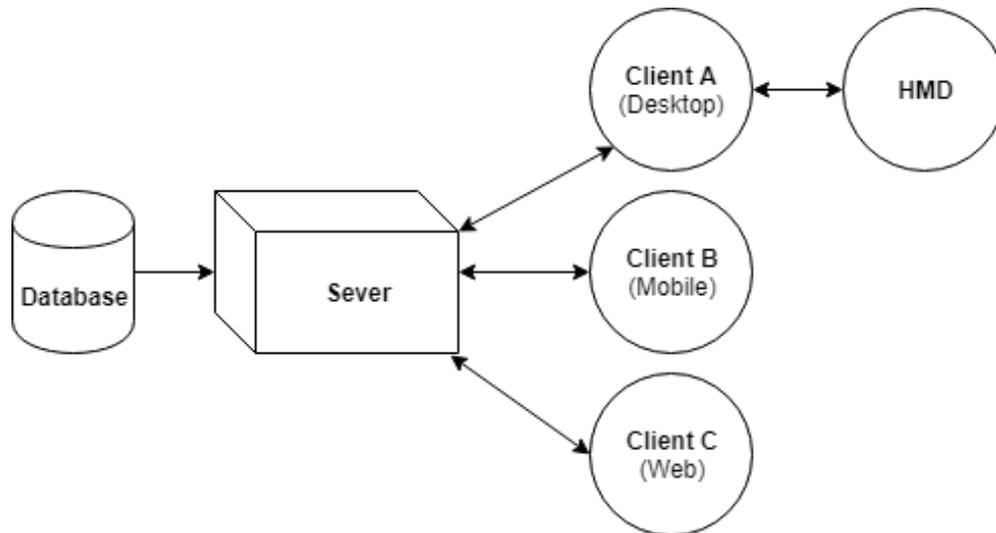


Figure 3: Architectural Diagram

2.5 Technologies Used

Our virtual reality application will be broken down into roughly 3 parts: front end asset creation, back end, and data storage in a database.

Graphics Engine

- **Unreal Engine 4** - is a complete suite of development tools made for anyone working with real-time technology. From enterprise applications and cinematic experiences to high-quality games across PC, console, mobile, VR and AR, Unreal Engine 4 will be our engine for our application.
- **C++** - is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation. It is used by the Unreal engine to build the core functionality of our system.

Assets

- **Maya** - an animation, modeling, simulation, and rendering software providing an integrated, powerful toolset. It can be used for animation, environments, motion graphics, virtual reality, and character creation.
- **Mixamo** - an online asset creation software by Adobe that allows for quick creation of 3D characters as well as creating animations for each asset.
- **Adobe Creative Suite** - is a software suite of graphic design, video editing, and web development applications developed by Adobe Systems. It will be used to create the artistic elements of our system.

Mobile Application

- **iOS / Swift** - is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod Touch. It will be used to create our mobile application which will host the audience mood toggle feature.

Web Application

- **HTML / CSS / JavaScript** - are markup and scripting languages predominantly used in creating web pages

and was used to create our web application. HTML and CSS were used to style the different web pages and JavaScript was used to drive the different events and actions off of the application.

- **React** - is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and corporations. **Redux** is an open-source JavaScript library for managing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Both of these libraries were used to make the user experience dynamic and responsive.

Version Control

- **Github** - is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management (SCM) functionality of Git as well as adding its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Server and Database

- **Node.js** - is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side. It will be used to communicate with our MySQL database and between our web.
- **MySQL** - is an open-source relational database management system. It will be used as our primary database for our system.

2.6 Design Rationale

This section will detail our overall usability and technology decisions for the application.

We chose this design in order to create a VR application usable with a HMD such as the Oculus Rift. The server and database in our architecture are important for supporting inputs into our application from a mobile platform.

The Unreal engine supports extremely realistic environments and uses the language C++, which all members of our team are familiar with. The alternative to the Unreal engine is the Unity engine, which uses C# and less realistic in our team's opinion. An advantage of Unity is a greater support for mobile than Unreal. However our initial platform is the Oculus Rift, and Unreal does have its own mobile port possibilities. We believe this makes the Unreal engine a better choice.

A risk we have identified in this project is asset creation for the virtual environment. To handle this, we have decided to use the tools Maya and Mixamo. Maya is a 3-D modeling tool recommended to us by Professor Max Sims based on his experience with 3-D modeling and Virtual Reality.

Maya supports creation and integration of 3-D assets into a game engine such as Unreal. Mixamo is an avatar creation tool that supports easy creation and animation of avatars, avatars sufficiently realistic for our requirements.

Lastly, in order to work on this project collaboratively, a version control system was necessary. GitHub was chosen because our team was familiar with the service and allowed multiple developers to work on the project at once. This not only helped with version control, but also helping track changes and catch bugs.

3 System Design for Virtual Reality Application

3.1 Overview

The primary component for our platform is the VR application, run on a system connected to a Head Mounted Display (HMD). We developed our system primarily using Oculus Rift, but expect it to function on an HTC Vive with minimal changes. Through the use of two hand-held controllers and the HMD, the user is able to view and interact with all virtual aspects of our platform. This section will provide detail regarding the functionality and implementation of each feature in the VR experience.

3.1.1 Activity Diagram

In beginning the VR experience, the user first is presented with a main menu where they can select to use our platform. From there, they are shown a variety of venues with a menu including the options to choose that venue, or preview the next. At the last venue, the choices cycle back to the first. Once the user chooses the venue, people are loaded into a room, and they are shown another pre-presentation menu. This menu includes a room code received from the server. At any point during the presentation, the user can use this room code to upload files into their presentation via the web client, or they can share it to a second, third, etc. user on the mobile client. During their presentation they can transition through slides and/or presentation notes using the two touch controllers. Finally at the end of the presentation they exit, and are taken back to the main menu screen.

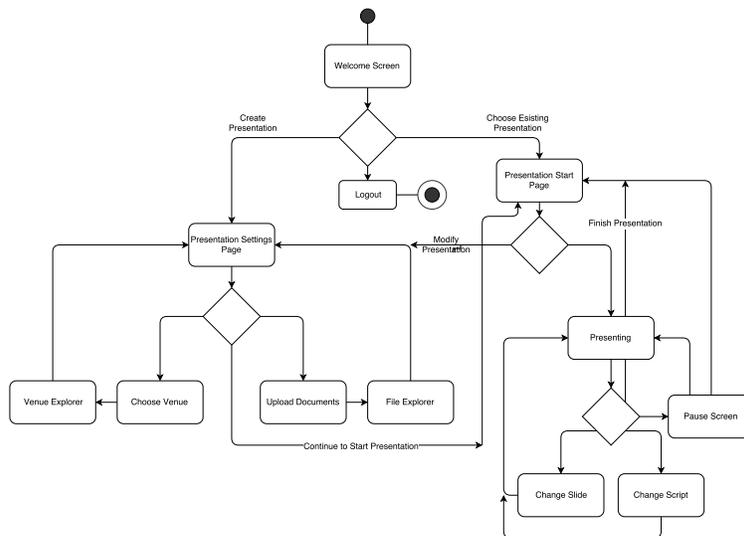


Figure 4: iOS Audience Control Page

3.2 Implementation

3.2.1 Asset Creation

In order for our system to simulate a realistic presentation environment, we needed to create assets to model the classroom and conference room we wanted to create. To do this, we utilized the help of Maya, and the Unreal engine. This section will walk through the process to create the assets that were used to model our two venues.

Classroom

Our goal when creating the classroom was to have the user feel like they were presenting in front of a class of students in a classroom that is similar to what they are used to. Therefore, we decided to model the classroom after a class we felt many engineers at Santa Clara University may have been in - **Engineering 326**. We found that the desks and chairs were purchased from the company Steelcase, and were able to find some models online. Using Maya, we were able to model textures and colors to replicate the chairs and tables seen in the classroom today - creating a realistic feel.

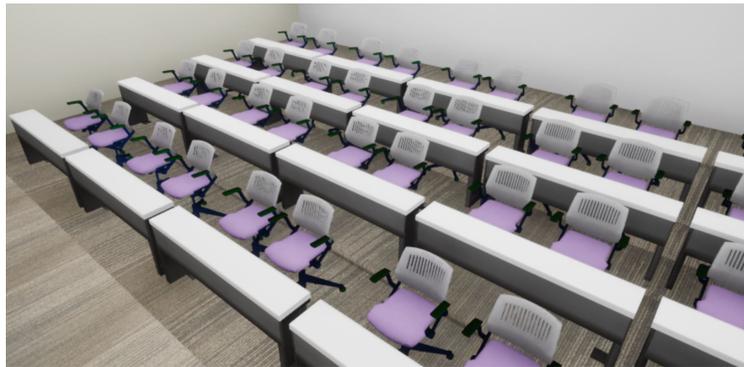


Figure 5: Classroom Assets modeled after Engineering 326

In addition to the tables and chairs, we were able to model the carpet by taking a picture and creating a material in Unreal that mapped to the image. Then by tiling it across the floor, our classroom model was able to mimic the carpet and flooring seen in Engineering 326. The Unreal engine was also very efficient in creating the walls and encasing structure of the classroom.

Conference Room

When designing our conference room, we wanted to model the environment off of a conference room you may see in a office building in a large city. As a result, we opted for glass walls, a long conference table, and office chairs to give a professional feel. Again we were able to use Steelcase's CAD designs to help create these assets and then add

textures and colors with the help of Maya. The glass walls were created in Unreal, along with the outside nighttime environment.



Figure 6: Conference Room Chair



Figure 7: Conference Room Table

3.2.2 Main Menu

Our main menu was designed in Unreal Engine 4 (UE4) using standard widget libraries. The menu is a 3D widget and serves as our applications title page in its own level. This first level provides the user two options, start presentation and exit. This level functions as the introduction to our platform, and is where the user will be taken when they finish their presentation. The option of Start Presentation takes the user to a venue select state, and the exit presentation will exit out of our application. The screenshot for this level is shown in figure 8.

3.2.3 Venue Select

The venue select functionality is implemented through transitioning the user into the different venue levels without the audience. The design decision of placing the user into the actual level was made based on the agreement that the



Figure 8: Main Menu

best way a user could preview a venue was to see it in 3D, rather than a photo preview for example. Included in the venue preview is a small widget providing two options, transition to the next preview, or select the current venue. This functionality was achieved through level transitions within UE4. Screenshots of our two current level preview options are shown in figures 9 and 10.

3.2.4 Presentation

The central part of VR experience is the actual experience of presenting in front of our virtual audience. The presentation level begins with a 3D menu introducing the user to the presentation. The menu shows the user a room code received from the server through a process described in section 3.2.7. At this point the audience members are talking with their neighbors, simulating a room as it would be before the audience is attentive. Once the user chooses the Start Presentation option, the menu changes into a countdown starting from 3 based on a timer. The start menu is shown below in figure 12. When the presentation begins, the audience state transitions to a presentation mode with randomized animations based on their interest and mood parameters (a detailed explanation of audience animation state is given in section 3.2.5). These parameters are set to default values at the beginning of the presentation but can be adjusted



Figure 9: Select Classroom

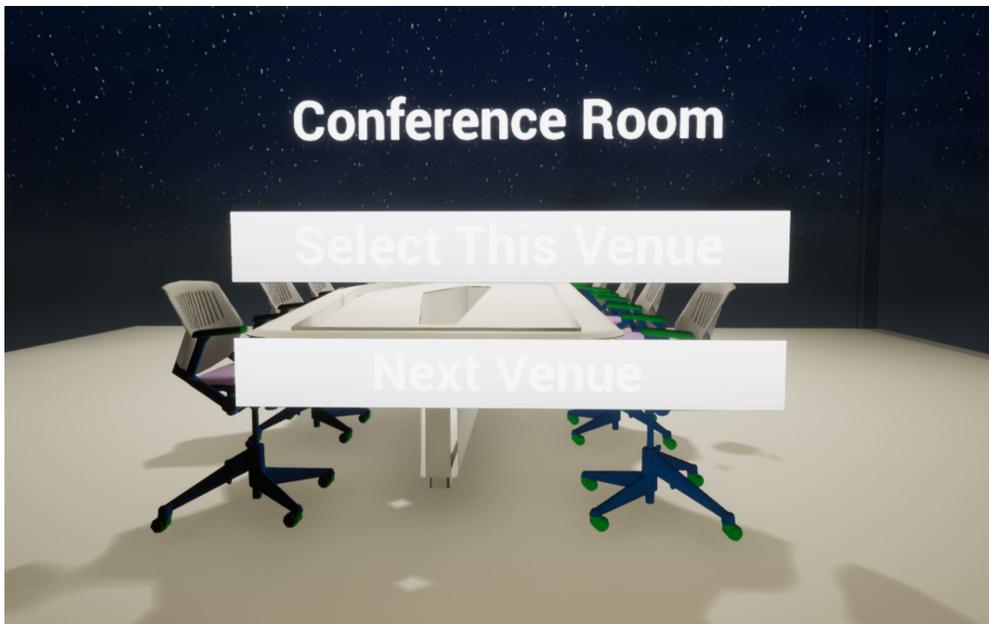


Figure 10: Select Conference Room

through the mobile client. With the idea of simulating real-life presentation clickers, the user is able to use the A and B (on Oculus) buttons on their right control to respectively increment and decrement the slide. The same is true for a notecard available to the user in their left hand. Pressing Y and X (on Oculus) respectively increments and decrements any notes they might have uploaded into the presentation. If no notes have been uploaded, no notecard is shown and the presentation screen behind them only shows "PresentVR". The slides and notes are loaded asynchronously from the server, a process described in more detail in section 3.2.7. By pressing an auxiliary button on the left or right

controller, the user is able to pause their presentation. In this pause screen they can see the room code, allowing them to at any point upload materials into their presentation or invite other players to participate in their presentation. At the end of their presentation, the user exits through this pause menu and is then taken to the Main Menu level.

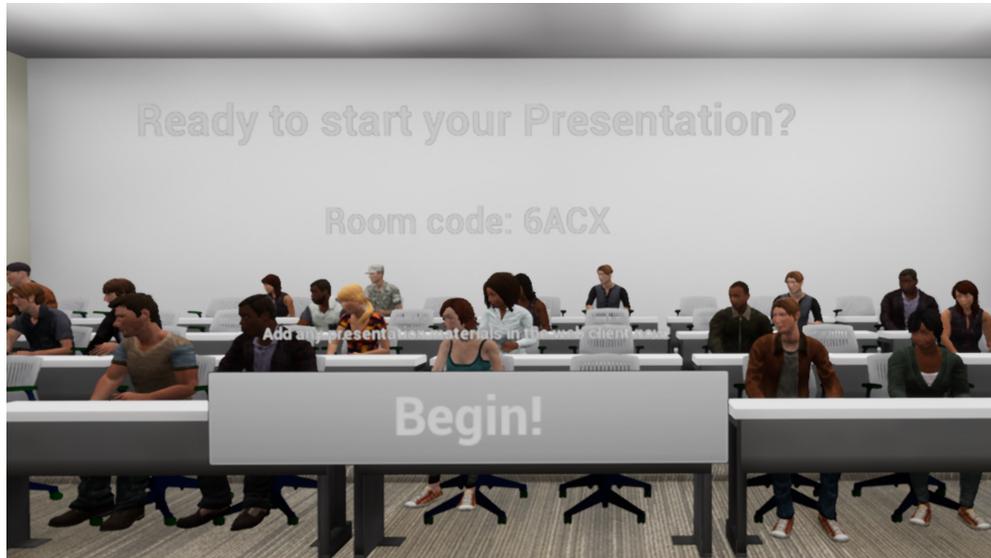


Figure 11: Start Presentation



Figure 12: Pause Presentation

3.2.5 Character Animation Model

In order to create as realistic an audience as possible, we created a complex character animation state model randomized but based on multiple parameters. For example, an audience member who is interested in the topic will have

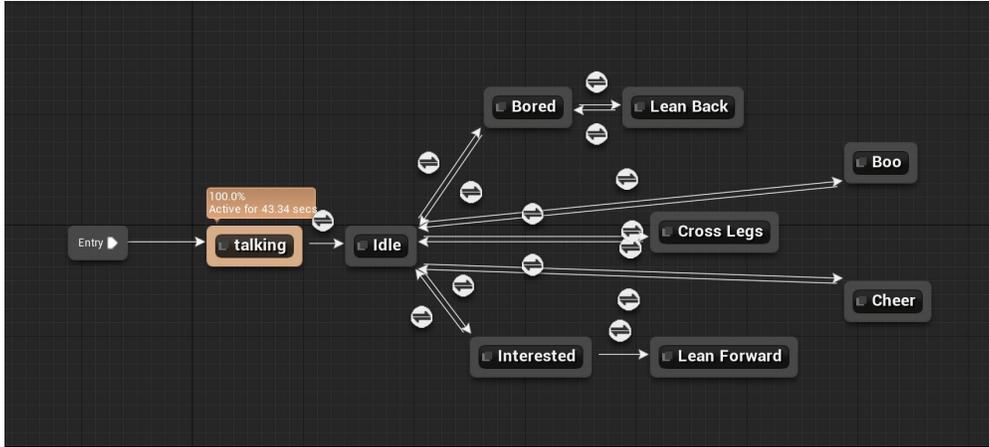


Figure 14: Audience State Machine

3.2.6 Presentation Aids

Presentation aids are one of the features of our platform that provide the most utility to the user. The ability to upload real-life presentations gives the user the ability to practice presentations they might have to give in the near future, and helping them prepare for a real-life scenario. More detail regarding the implementation of this functionality in the web platform is given in section 4. In the VR platform, the slides are requested from the server using the room code and current slide number. The notes are requested in a similar way, and exist as a line of text. As mentioned above the interface to move through these aids is simple and mirrors the real-life functionality of a presentation clicker. The current place in the presentation slides and notes are stored in the user as a variable, and when the user increments/decrements the respective aid, the corresponding variable used in the request is changed, and finally a request made.

3.2.7 Server and Communication

Our VR Application communicates with our other platforms through a server written in NodeJS. It is through this server and a corresponding database that all information is stored and remains persistent. The VR application communicates with the server through a socket connection. We used an open-source socket plugin for UE4 to implement much of the functionality necessary in our application. A connection is first made after a user chooses a venue. At the same time the user is presented with the Start Presentation menu, the level requests a socket connection on a specific port through the plugin mentioned previously. Once the connection is created, the server uses the socket id to create a room code. It first saves the socket id and corresponding room code to the database, and then sends the room code back as a response to let the application know its successfully connected. The level saves this code and triggers an

update on the Start Presentation menu to display the new room code.

The socket connection is kept open during the presentation, and is utilized by the server to update character emotions. Once the server receives a cue to update the audience emotions in a certain room, the server finds the corresponding socket id through a database API, and then sends an emotion update message to a given socket. The VR client parses these messages, and updates the audience emotion parameters correspondingly.

The third functionality the VR client makes use of with the server for are the presentation aids. When materials are uploaded through the web client to a specific room, the server sends a message through the socket to alert the application materials are available. Retrieving the content of the aids only required one-way communication, as the user would be the only one deciding when to increment the presentation materials. Combined with the complexity of receiving files through the socket plugin, we opted to implement a RESTful API instead for this functionality. To retrieve slides, the application makes a GET request to the server URL with room code and slide number as query parameters. The server parses this request, uses the database API to find the correct file, and then sends it in response. The application takes this file, creates a new material out of it and then finally assigns the material to the presentation screen behind the user. A similar flow is used for retrieving the presentation notes. The application makes a GET request providing the note index and number of words as query parameters. The server through the database API finds the correct place in the note text, and sends the response back as a JSON message. The application then takes this message and assigns it to a text widget in the users left hand. Once the user completes their presentation and exits, the server receives a socket disconnect message, and deletes the room record from the database.

4 System Design for Web Application

4.1 Overview

Our platform utilizes a web application in order to upload presentation materials into the VR client. We chose a web-based client for this application in order to simplify the user experience between presentation creation and uploading. Presentations are typically created in PowerPoint or Google Slides and then exported locally. Our web application provides a convenient interface to upload presentation materials, save presentations to their account, and practice their presentation at any point in the future.

4.1.1 Activity Diagram

When using this feature, the user first has two options. They can either create a new presentation, or choose to use an existing one. The new presentation flow assumes that the user already has exported their desired presentation, and has a text file containing their presentation aid notes. In the web interface, the user can select Create a New Presentation, with which they are presented a blank presentation form. There they can name their presentation, upload materials, and save it. Once saved they are taken back to the main menu, where they then input their VR room code into the designated form, and can return to the VR platform. If the user chose to use an existing presentation (one they had previously created and saved), they could simply enter the room code on the dashboard screen. The relevant UIs are shown in the following section.

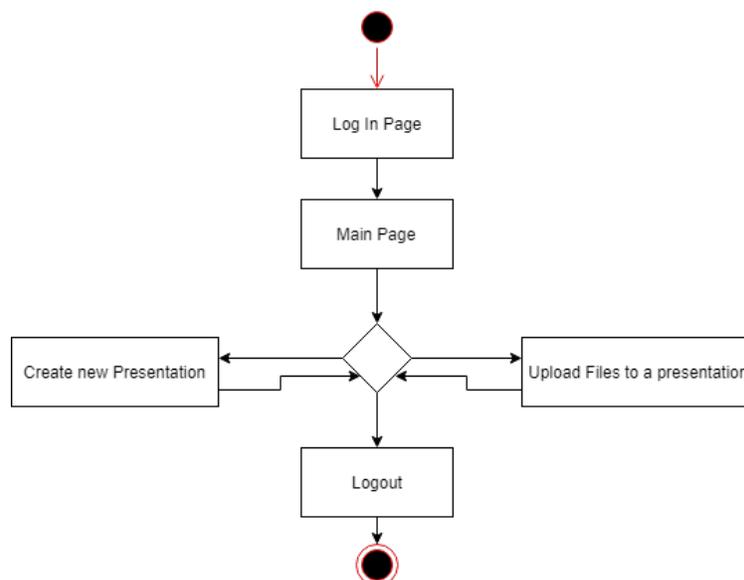


Figure 15: Web Activity Diagram

4.2 Implementation

4.2.1 User Interface

The user interface of our application was implemented using the React Javascript library. We wanted our application to be responsive and interactive, decided React was the best choice. In order to manage data for users and control the state of our application we chose to use Redux as well. Our team had experience with both React and Redux which we factored into this decision. The primary screen of our UI is the dashboard show in figure 16. This screen shows the user all of their previously saved presentations. It also gives the user the options to create a new presentation or log out. The create presentation option takes the user to the form shown in figure 17. With this form the user can upload their presentations aids. The main menu screen assumes that the user is logged in. If they are not, they are initially directed to a landing page and from there can login (figure 18).

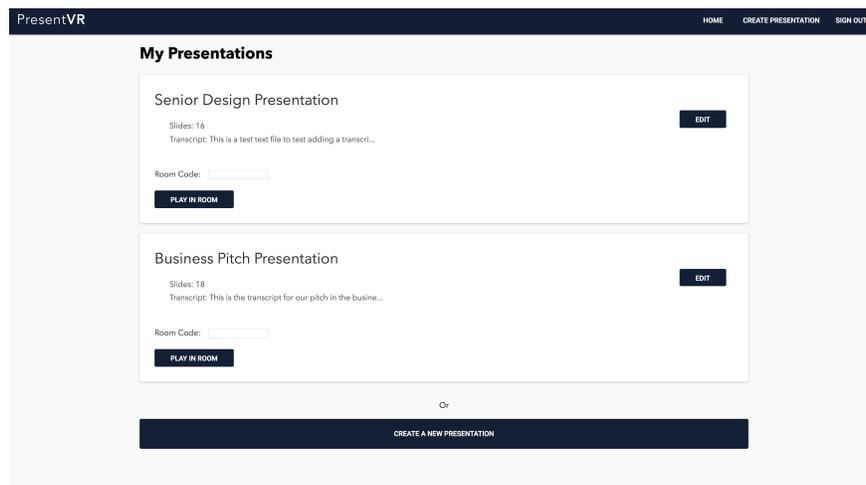


Figure 16: Web Dashboard UI

4.2.2 Application State Model

While React is great for creating user interfaces, it doesn't have state requirements necessary for our fully functioning web app. Redux is a predictable state container for Javascript apps, and gives the necessary functionality. Our application state is modified with four different actions: login, load presentation, create presentation, and add presentation to room. Using this data we can show users which presentations they have and which presentations currently correspond to which room. React and Redux make the user interface interactive while limiting API calls.

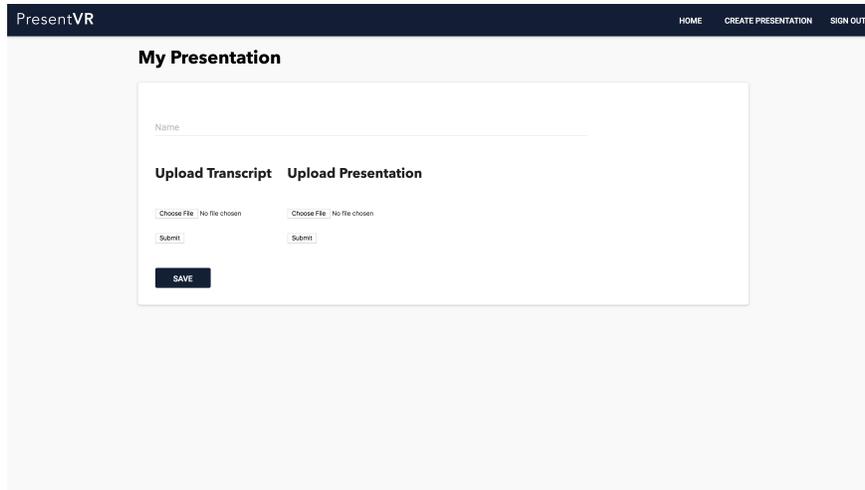


Figure 17: Web Create Presentation UI

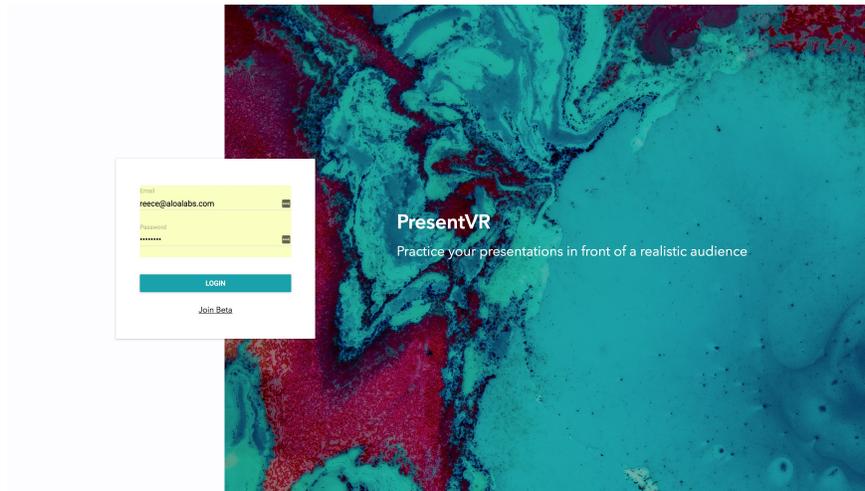


Figure 18: Web Login UI

4.2.3 Server and Communication

The web platform communicates with our server through a REST API. A full list of relevant endpoints is given in appendix A. The API gives the web client the ability to create new users, log in existing users, get presentations for a user, create new presentations, etc. All of these actions have a corresponding server-database interface to keep this information consistent. Files uploaded to the server are linked through a corresponding presentation id, which then has a specific user id.

5 System Design for Mobile Application

5.1 Overview

One main goal in the development of our project was to create as realistic an experience for the primary user. One of the challenges our team identified in this process was having the audience respond to the user in a believable way. We solved this problem with a multiplayer experience through the creation of a mobile iOS application that allows third party control of the presentation audience. Based upon the content or quality of the presentation another user is able to input the room code on their mobile device, and change the audiences behavior parameters. The mobile user is also able to trigger more dramatic actions such as applaud or boo, whether they want to support or heckle the primary user.

5.1.1 Activity Diagram

The mobile use case is the simplest in this project. After pulling up the application on their mobile phone, the user is able to put in the room code from the presentation in the VR client. If it is valid, the user is taken to a screen with two sliders for behavior parameters, and two actions they can trigger. From there the user can interact with these components or leave the room.

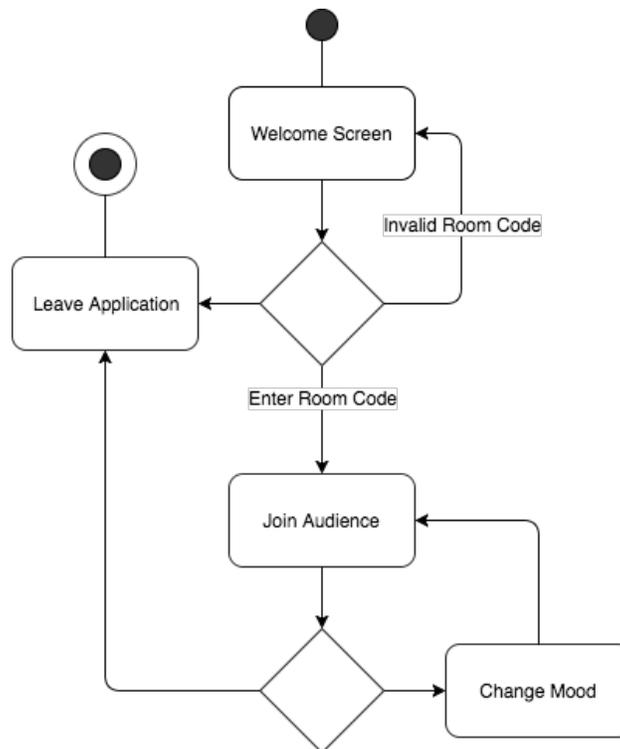


Figure 19: Mobile Activity Diagram

5.2 Implementation

5.2.1 User Interface

The user interface on this platform was designed to be as intuitive as possible. The first page allowing the user to input the room consists of only one form. The input is automatically capitalized and limited to 4 characters to make format consistent with room codes. This screen is shown below in 20. The second screen in the application likewise has a very clear and simple UI that lets the user change either the interest level or mood, or trigger the applaud or boo actions.

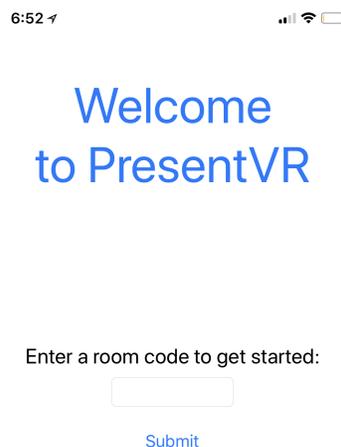


Figure 20: iOS Enter Room Code

5.2.2 Server and Communication

The iOS application communicates with the server through HTTP requests using the servers RESTful API. When the user submits the room code, the application sends a request to make sure the room code is valid. The server responds with the socket id, saved in the applications state. When the user updates the mood or interest parameters in the app, a post request is sent to the server with the new emotion value and the socket id. The server then sends the parameter

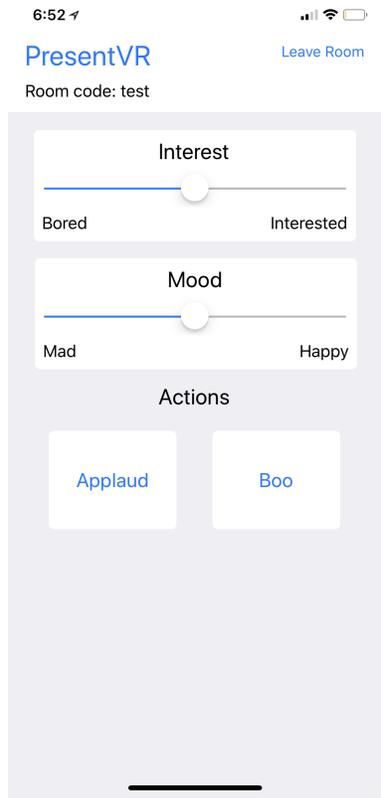


Figure 21: iOS Audience Control Page

label with the new value to the specified socket, which is then interpreted by the VR platform.

6 Testing and Results

To ensure that our system was functioning the way we intended to, testing was necessary. By testing at each phase of the development process, we were able to catch errors earlier on as well as receive feedback that helped us improve our system.

6.1 Alpha Testing

Testing begins with Alpha testing, where we verify that the internal functionality of the system is working. This is done through white box (internal functionality is known) testing in which we used unit, integration and regression testing. These tests were run throughout the development process and across our multiple platforms to ensure that the functionality was working from the code perspective. Unit testing refers to tests that are run on individual parts of our system, or smaller units, to ensure that each individual component was working. Our team would test each button, action, and process within each level of the virtual reality application, each page on the web application, and on the mobile application.

Additionally, integration testing was one of the most important steps during the testing process. During this phase, we connected or integrated each individual component to ensure that the system functioned as a whole. The components were joined one at a time so that we could pinpoint where a bug or issue would come from if it arose. We first connected each blueprint in the virtual reality application and then each level. Once that was complete, we began to link the web pages and ensure that the website worked smoothly. And then lastly, we linked the different pages on the mobile application. Once each part of the overall system was functioning, we integrated them together to test if data was properly being transferred, sessions were being created, and the audience mood was changing. Regression testing is similar to integration testing, but it helped to make sure that once we added something to the project, that everything that was previously there did not generate any bugs or errors.

6.2 Beta Testing

The second phase of testing is Beta Testing, where once the system is operational, we had volunteers test out our project to help provide feedback for improvements in design and interface. This is also where we did Black Box testing, where the user does not know the functionality of the system and what it is supposed to do. Through a virtual reality class, we saw that there was a live demo session in which students would display their projects and have random users give feedback. We attended this workshop as well as asked friends and colleagues to try out our application.

From the feedback, we then changed aspects of the system that were repeatedly brought up as well as we agreed would help create a better application. Some suggestions were:

- To not have the blueprints and widgets be locked in relation to the head mounted display as it induced dizziness and made the application uncomfortable to use.
- Needed to make an executable in order to eliminate loading time and poor latency while using the project.
- Characters were not as realistic and lacked the proper textures to feel like they were realistic.
- Volume levels and sounds in the application were too loud.

By using beta testing we were able to improve the usability of our system and catch issues that from a programming point of view, were not able to be seen.

7 Issues and Constraints

Today, it has become very important to understand the many different issues that can come forward when programming and working with content that is available online. Some issues we had prepared for ahead of time, while other issues arose while working on the project.

7.1 Ethical

Our team recognized that there might be some ethical conflicts while working on this project as we used open source libraries as well as used some CAD designs we found from companies like Steelcase online. While looking at these resources, we needed to ensure that they were free to use or were under the Creative Commons License(CCL). The CCL allowed our team to use other people's work legally. Additionally, it was also important to not charge money to use our application as we had used work from others for free, and should not make money off of it. We also were managing user passwords and information, and wanted to make sure that it was securely encrypted and safe. To do this, we used Google's Firebase service to keep the data safe and secure.

7.2 Social

As a result of our project, we hope that the social context around public speaking and anxiety as a result of public speaking begin to change. Our system will be free to the public use and will allow people across the world to have access to a tool that will help them become better speakers.

7.3 Political

Our team believes that if our system is successful, that it may be helpful to politicians in their effort to become better speakers to a wider audience. Other than that, we don't believe that our project is very political in nature.

7.4 Economic

So far, our project has been completely free to develop. With the help of the VR lab on campus, there haven't been any costs associated with us. Additionally, we plan to release the application free to the user, reducing the costs for a customer to use the product. Additionally, with the advances in technology, we hope that the price to acquire a virtual

reality headset decreases so that the only cost to the user (the headset) is significantly reduced. If economic issues arise in the future, we hope to bootstrap the project to raise money.

7.5 Health and Safety

Our project is a Virtual reality project and it was important to consider different health and safety issues that could arise from using a head mounted display(HMD). For instance, we found that a previous version of our system caused users to get dizzy, so we needed to account for the user experience when a presenter was using the system. Additionally, it was important for us to ensure that the user did not hurt themselves while using the application. As a result we had warning messages to make sure their surroundings were clear and that they would not hit themselves on anything while using the application.

7.6 Manufacturability

Our project has no physical components associated to the actual application. The user must own a virtual reality headset, but there aren't any production issues from our application.

7.7 Sustainability

When looking at our product, a main issue with regard to sustainability is that once a user thinks they are a "great" public speaker, they will stop using the application. As a result, we hope to add gamification and other additional features that would entice users to come back and use the application. We also hope to form partnerships with school districts where our product can be used to help students directly in their educational environments.

7.8 Environmental Impact

Since our project is software based, there isn't any physical production required and as a result not much environmental impact. However, if demand rises, the demand for virtual reality headsets will also rise, meaning that whatever environmental impact is currently produced from the creation of these devices will also increase. To combat this, we will try and lobby virtual reality companies to start making their products more sustainable when the environmental impact becomes more of an issue.

7.9 Usability

A main issue our team worked hard to try and solve throughout this process was to make sure that the user experience was friendly and easy. We made sure that the flow was simple and that the UI / UX was intuitive for the user. A main issue we ran into was uploading presentation aids via the web application, as it broke away from completely experiencing the application in the head mounted display. However, we looked into the file system manager within the Unreal Engine and found that process to be much more convoluted and less friendly. As a result, we made the web application straightforward and visually appealing.

7.10 Lifelong Learning

For this project, we decided to tackle a field that was unknown to us: virtual reality. This project has motivated us to further our knowledge with virtual reality applications and explore similar fields such as augmented reality. These technologies are both relatively very new and still have a lot of research and development going into them. We also gained experience with graphic and three-dimensional design, an area none of us had experience with. These were all important skills to have learned and can be applied in future projects later on in life.

7.11 Compassion

Our project aims to help those who face anxiety in their everyday lives that is associated with public speaking. We hope that their suffering can be mitigated or resolved with our application. As Santa Clara University students, we were taught to lead with compassion, and it was a central motivating factor when coming up with which problem we wanted to solve for our senior design project.

7.12 Risk Analysis

This section describes the possible risks that were associated with this project after we had come up with our requirements and design decisions. The Risk table below shows our predicted risks as well as the likelihood and severity of these risks, the potential consequences, and mitigation strategies. The total impact of the risk is calculated by multiplying the severity with the probability.

Risk	Consequences	Probability	Severity	Impact	Mitigation
Unfamiliar Programming Languages / Technologies	Time spent learning the new language, mistakes during development	0.8	6	4.8	Have different team members learn a different technology fluently.
Time	Entire System not being finished	0.5	6	3.0	Set deadlines, have a SCRUM master.
Team getting sick	Project delayed	0.5	5	2.5	Emergen-c, proper rest.
Minor bugs	Project delayed	0.5	0.4	0.2	Test.
Major bugs	Application issues	0.1	4	0.4	Leave ample time for testing.
Data loss	Project delayed	.01	5	.05	Use version control.

Table 1: Risk Analysis Table

8 Conclusion

8.1 Summary

We believe that our overall system was able to effectively solve the problem we set out to address: the lack of a proper platform to practice public speaking conveniently and easily. As described in this document, our system provides the user the ability to practice a presentation in different venues while having an audience that feels real and responds throughout the presentation. That being said, we faced a lot of challenges, made many mistakes, but in the end learned a lot and made a system that we have enjoyed using.

Throughout this process, we were able to create a virtual reality application using the Unreal Engine that made the user feel like they were in front of a live audience. We were able to develop a web application that was connected to our Node server, with the ability to allow users to upload their presentation materials and have that data be sent to our virtual reality application. Presenters could click through their slides and look through notes cards just as they would in real life. In addition, our mobile application did a great job at adding multiple users to the experience in which they were able to change the mood of the audience and help create a more realistic experience.

Along the way, we faced many challenges and obstacles that sometimes felt could have derailed the entire project. We had very little experience with virtual reality, busy schedules and severe personal circumstances, as well as many complex bugs and issues we had not anticipated. Yet, we found a way to overcome them and create a project that we are all very proud of. Overall, this experience has been very rewarding and fun. We have been exposed to many new technologies that we have previously not know and are looking forward to seeing how this application can be improved upon and useful in the future.

8.2 Lessons Learned

During the last nine months there have been many obstacles that our team has faced and many lessons that we learned from it. Some are:

- Graphical design is no joke and can be very time consuming. Our team comprised of all computer engineering majors and many of us had little to no previous experience with graphical design and in particular, three-dimensional design for virtual reality.
- Scoping is a very important aspect and it is important to be realistic with how much you think you can get done within a given time frame. When coming up with what we wanted to do initially, our idea comprised of many small additional features that once were put in scope, made the project unrealistic to complete within the time given. Especially with the ever-changing schedules of college students, there wasn't much time to complete all that we wanted to do. Additionally, we would add features along the way that not only took more time, but also brought unforeseen complexity issues along the way.

Through these lessons, we have taken away key value that will stick with us throughout our lifetime and will be better prepared to avoid and recognize problems in the future.

8.3 Future Potential

Over the course of senior design, we came to realize how effective and useful this application can really be. After building our initial system, we believe that we have established a great baseline that can be built off to create an application that people not only will want to use, but can become better public speakers through.

Analyzing the User

We believe that adding analyzing the user and using that feedback in our system could really improve the quality of the experience as well as more realistic. One idea is to use a smart watch to analyze the user's attributes such as heart rate to determine when they might be getting more anxious, and to adjust the audiences reactions based off of this metric, or see what audience actions generated this reaction by the presenter. This data could help the user see what causes them to slip up, speak faster, or become anxious and address ways to help solve the problem. Additionally, we also aim to take in the speech of the presenter to analyze the words being said as well as the tone of their voice. The tone in combination with the heart rate can provide more accurate analysis to the state of the presenter. Analyzing the words being said could help gauge whether or not they are speaking too quickly, or how many filler words they say

(such as umm, or uhh). Using this data can again, help the user see what they can do to improve their presentation. Additionally, we believe that implementing a dashboard that can be seen at the end of the presentation on the web application can help the presenter see a visual representation of how they can improve their presentation and become a better public speaker.

Different Modes

We would also like to add different modes to our application in the future. For example, a challenge mode or mini-game that a user can choose to play to see if they can present against the toughest of audiences. The audiences would heckle, boo, and make it very difficult for the presenter to stay on top of their speech and not get distracted. Modes like this would entice the user to use the application as well as make it a more fun experience. We could also add gamification to the application where we could score presentations and users could compete against their friends to see who is the best public speaker.

Appendices

A Server REST API Reference

A.1 UE4 Endpoints

Method	Endpoint	Request Params	Body	Query Params	Response
GET	/r/:room/slides	Room code	-	index data (optional - default false) download (optional - default true)	file
GET	/r/:room/transcript	Room code	-	words index	{ text: "string" }

Table 2: UE4 REST Endpoints

A.2 Web Platform Endpoints

Method	Endpoint	Request Params	Body	Query Params	Response
GET	/presentations/:userId	User Id	-	-	{ presentations: [...] }
GET	/login	-	-	username password	{ user: { ... } }
POST	/newUser	-	username password	-	{ user: { ... } }
POST	/p/:presentationId/slides	Presentation Id	files	userId	{ "success": true }
POST	/p/:presentationId/transcript	Presentation Id	file	userId	{ "success": true }

Table 3: Web REST Endpoints

A.3 Mobile Client Endpoints

Method	Endpoint	Request Params	Body	Query Params	Response
GET	/r/:room	Room code	-	-	{ sessionId: "sessionId" }
POST	/r/:sessionId/interest	Session Id	-	value	{ "success": true }
POST	/r/:sessionId/interest	Session Id	-	value	{ "success": true }
POST	/r/:sessionId/interest	Session Id	-	value	{ "success": true }

Table 4: Mobile REST Endpoints