

---

# **Administrator Manual**

## **for**

# **Virtual Video Modeling on the Social Skills of Adults with Autism**

**Version 1.0 approved**

**For Sarah K. Howorth**

**University of Maine**

**April 15, 2024**

**Prepared by JamTech:**

**Tristan Cilley, Allison Lupien, Nick Sarno,**

**Jacob Michaud, Maha Fazli**



# **Virtual Video Modeling on the Social Skills of Adults with Autism**

## **Administrator Manual**

### **Table of Contents**

	<u>Page</u>
1. Introduction.....	3
1.1 Purpose of This Document.....	3
References.....	3
2. System Overview.....	4
2.1 Background.....	4
2.2 Hardware and Software Requirements.....	4
3. Administrative Procedures.....	5
3.1 Installation.....	5
3.2 Routine Tasks.....	6
3.3 Periodic Administration.....	6
3.4 User Support.....	8
4. Troubleshooting.....	9
4.1 Dealing with Error Messages and Failures.....	9
4.2 Known Bugs and Limitations.....	9
Appendix A – Team Review Sign-off.....	11
Appendix B – Document Contributions.....	12

# 1. Introduction

This is a project for Dr. Sarah Howorth on virtual video modeling of the social skills of adults with autism. Dr. Howorth is the director of the PEERS® Lab at UMaine and our primary client for this capstone project. The PEERS® curriculum, a formal process for teaching social skills, was written by Dr. Elizabeth Laugeson. Our client's project proposal outlined a request to make the curriculum available in a virtual reality environment.

The JamTech team consists of five students currently enrolled in a two-semester (Fall 2023 - Spring 2024) computer science capstone course at the University of Maine. Each member of the team has volunteered for a specific team role which allows us to delegate assignments to everyone on the team.

## 1.1 Purpose of This Document

The intended audience is Dr. Howorth and other researchers. The purpose of this document is to inform administrators on how to run and maintain the application in order to use this application as a research tool. It gives an overview of the hardware and software requirements, how to install the application, common tasks that need to be performed, where to find user support, and how to troubleshoot common issues.

## References

- PEERS®. (2021). PEERS® (version 1.1.0) [Mobile app]. Apple Store OR Google Play. [https://play.google.com/store/apps/details?id=com.peersclinic.peers&hl=en\\_US&gl=US](https://play.google.com/store/apps/details?id=com.peersclinic.peers&hl=en_US&gl=US)
- PEERS (2023) UCLA PEERS® Clinic, Semel Institute for Neuroscience and Human Behavior. <https://www.semel.ucla.edu/peers>
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2024). GitHub repository, [https://github.com/VoloVita/PeersVR\\_Capstone/tree/main/Documentation/Deliverables](https://github.com/VoloVita/PeersVR_Capstone/tree/main/Documentation/Deliverables)
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). System Design Document (SDD), version 1.0
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2023). Systems Requirement Specification (SRS), version 2.0
- Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud, Maha Fazli. (2024). User Guide (UG), in progress

## 2. System Overview

This is an overview of the system for the system administrator. The following sections summarize daily operations and maintenance, and lists the technical requirements of operation.

### 2.1 Background

The system administrator is responsible for acquiring the source code, installing the application, performing routine administrative tasks, and providing user support. There are specific guides on each of the administrator's responsibilities including forking the PeersVR\_Capstone Github repository, building the .apk from the source code, and loading the .apk onto an android based headset. The PEERS® VR program does not require any routine maintenance. Additionally, administrative tasks may also include expanding or adding content to lessons. These tasks are explained in more detail in section 3: Administrative Procedures.

### 2.2 Hardware and Software Requirements

In order to run and update the application, there are specific hardware and software requirements that need to be met depending on the mode of execution.

#### **Running the Application on a VR Headset:**

- The application was designed for, and tested on, the Oculus Quest 1 headset, but it should be compatible with any version of the Oculus Quest or newer.

#### **Running the Application in Unity w/o Headset:**

- Windows Operating System (recommended)
- Unity Editor version *2021.3.34f1* (required)
- Dedicated GPU (recommended)
- Unity XR Simulator Plugin (recommended)
- Unity XR Interactive ToolKit (recommended)

#### **Running the Application in Unity w/ Headset via Quest Link:**

- Windows Operating System (recommended)
- Unity Editor version *2021.3.34f1* (required)
- Dedicated GPU (required)
- USB-C cable (recommended)

### 3. Administrative Procedures

The administrator procedures will assist users in both installing and maintaining the system, while also providing access to user support as required.

#### 3.1 Installation

The following section explains the process by which a new administrator can acquire the source code, build the project, and load it onto a VR headset. GitHub has detailed [documentation](#) for the use of their platform, we recommend that anyone unfamiliar with GitHub should familiarize themselves with basic operations before continuing.

##### Acquiring the Source Code:

The source code for the PEERS® VR program can be acquired from the [GitHub repository](#). It is recommended that the new administrator forks this repository to ensure that they possess an identical copy of the project without being dependent upon the availability of the original repository. Once the project has been forked, the new administrator should clone the repository. This will create a copy of the project's source code in the administrator's local environment. The entirety of the source code will be contained inside a folder called "*PeersVR\_Capstone*".

##### Opening the Unity Project:

In order to view the project inside the Unity Development Platform, the new administrator must have a Unity account. [Sign-up](#) for a Unity account.

Additionally, the administrator must also have [Unity Hub](#) installed to open and manage Unity Projects. PEERS® VR was developed using the Unity editor version *2021.3.34f*. This unity installation is required in order to open the project from Unity Hub. Unity editor version *2021.3.34f* can be installed through Unity Hub, or from the Unity website. To view the project inside Unity, open Unity Hub and add a project. This action should prompt the administrator to specify where the project files are located. As mentioned previously, the project will be contained inside a folder called "*PeersVR\_Capstone*".

To reduce the size of the GitHub repository, the project folder only contains essential files. This means that Unity will have to create additional folders such as Library, Build, and User Settings. This process only occurs once and will happen automatically the first time the project is opened from Unity Hub. Due to the size of additional folders needed, it may take a few minutes.

## **Acquiring the PEERS® Video Content:**

The PEERS® VR program contains media content in the form of example roleplay videos and thumbnail images. Since this content is private property, it is not included in the public GitHub repository and must be downloaded separately. JamTech will provide this content to the new administrator via a private folder on Google Drive. This folder must be downloaded and placed inside the JamTech\_assets folder at: *PeersVR\_Capstone/Assets/JamTech\_Assets*

## **Building an .apk from source code**

Following [this YouTube tutorial](#), the administrator will be able to download the appropriate android SDK and JPK addons for Unity in order to be able to build the executable .apk(android/VR) file. After downloading and installing these, if they continue following the instructions of the tutorial they will set up and optimize the project for the Oculus device via build settings and project settings in Unity. Lastly, when these settings are completed and they are ready to create an executable, they press the build button and set a local location on their computer which it will be stored in.

## **Loading the .apk onto headset:**

In order to be able to load the .apk file onto a headset standalone, you will need to have access to a developer account for Unity, which essentially enables you to put non licenced applications that you have created or others onto the device. The account you use does not have to have special permissions other than being a developer account. Once you have access to and are logged into the developer account on your headset, download the [SideQuest application](#). If you have created and verified the unity account, this app will guide you to setting up the device and downloading the app onto it. Once you download the app onto the oculus, go into apps→unknown sources→the name of the app you built. This will run the application on the headset.

## **3.2 Routine Tasks**

For this project, there aren't many maintenance tasks that need to be done. If for any reason there is a bug during runtime it is recommended to restart the application.

## **3.3 Periodic Administration**

The PEERS® VR program does not require regular maintenance, but it is possible to expand the content of the program to include more of the PEERS® curriculum.

## Adding Additional Lesson Content:

As of the conclusion of the Capstone course, the first five lessons of the PEERS® curriculum are fully functional. It is possible to expand the program to include more lessons, but it will require additional labor. Each additional lesson will require the creation of a new folder (named: topic\_<lesson#>) inside the PeersVideosContent directory. Inside the topic folder two more folders will be created, one for video files (mp4) and another for thumbnail images (jpg). The .meta files are created automatically by Unity once the project is opened.

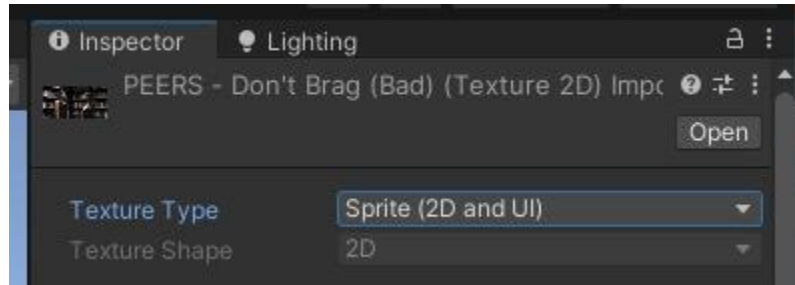


**Figure 1.** Lesson Content: *Each lesson requires a folder of necessary files, including thumbnails and videos.*

The program stores the information for all lessons inside a JSON file at: *PeersVR\_Capstone/Assets/JamTech\_Assets/JSON/lesson.json*. This file will need to be expanded to include the text information of the new lesson (title, description, etc) as well as the relative file paths to the new video and thumbnail images. It is absolutely vital that the file paths match the names of the files

The program stores the information for all quizzes inside a JSON file at: *PeersVR\_Capstone/Assets/JamTech\_Assets/JSON/quiz.json*. This quiz file will also need to be expanded to include the text information of quiz questions associated with each new lesson.

Note: When adding new thumbnail content to a unity project, it will be necessary to set the texture type of the file to 'Sprite (2D and UI)' inside the inspector panel (as seen in Figure 2 below)



**Figure 2.** 2D Sprite: *Images should be turned into 2D Sprites by changing the Texture Type.*

Additionally, new ReadMore panel content will need to be added to the main scene of the project (as seen in Figure 3 below).



**Figure 3.** Lesson Objects in Scene Hierarchy: *Each lesson needs an object under the ReadMore panel, as seen in the scene hierarchy view.*

### 3.4 User Support

To acquire user support for this document, please refer to these specified deliverables on the application GitHub:

- **JamTech\_AM** : which is the Administrator Manual, the manual contains all the information needed for installation, troubleshooting, etc..
- **JamTech\_UG** : which is the User Guide, maybe a quick description of pdf, this document's purpose will be for any guidance that is intended for the team or developers.



For further details and assistance, kindly reach out to us via the team email address:  
[JamTech397@gmail.com](mailto:JamTech397@gmail.com)

## 4. Troubleshooting

This section will cover troubleshooting methods and responses to error messages, failures, or known bugs within our program, including what they are and how to deal with them if or when they occur.

### 4.1 Dealing with Error Messages and Failures

To handle errors and failures while expanding, the administrator will need to perform manual testing. Manual testing can be performed by entering play mode in Unity, interacting with the scene and the elements under development. The developer can then change elements in play mode to see realtime effects. It is important to note that changes made in play mode will not be saved. To save changes, move back into edit mode, make the necessary changes and save. The new changes can be manually tested by entering back into play mode.

If an error of failures occurs while using the current application, the user should restart the application.

### 4.2 Known Bugs and Limitations

- Summary:** When scrolling the content of the lesson view vertically, it is possible for the cursor to get stuck in the horizontally-scrolling content containing example and non-example videos which prevents the user from scrolling vertically.

**Location:** Lesson View Panel

**Solution:** We can add colored backgrounds to the scrollable areas to help users distinguish between them.
- Summary:** The quiz view does not provide feedback for correctness of responses like it should.

**Location:** Quiz View Panel

**Solution:** We can add the visual pop up which informs the user if their quiz response was correct or not.
- Summary:** The curriculum view needs a second step for selecting a lesson to display which lesson is being selected.

**Location:** Curriculum View Panel

**Solution:** Selecting a lesson from the curriculum panel should be one action.  
Transitioning from the curriculum view to the lesson view should be a separate action.

4. **Summary:** When building the executable, the lesson content does not load to the lesson view when selected.

**Location:** Discrepancy between script elements and inspector details.

**Solution:** It is possible to assign object references in the inspector panel by executing a script. This will ensure that the program builds successfully.

5. **Summary:** Only the first 5 lessons are implemented so far; the rest will redirect the user to the first lesson.

**Location:** lesson.json, quiz.json, and PeersVideosContent

**Solution:** For more details, see section 3.3: Periodic Administration.

6. **Summary:** No UI element programmed to exit the application.

**Location:** N/A

**Solution:** The program either needs to be stopped in the Unity Editor, or closed in the VR headset.

## Appendix A – Team Review Sign-off

This is the team review sign off meaning that all current team members of JamTech (Tristan Cilley, Allison Lupien, Nick Sarno, Jacob Michaud and Maha Fazli ) have fully reviewed and read the administrator manual and do agree with the content and format included in the document.

By typing one's name under the signature column and giving the date, the individual signs this document.

Name	Signature	Date
Allison Lupien	<i>Allison Lupien</i>	04/15/24
Comments:		
Jacob Michaud	<i>Jacob Michaud</i>	04/15/24
Comments:		
Maha Fazli	<i>Maha Fazli</i>	04/15/24
Comments:		
Nick Sarno	<i>Nick Sarno</i>	04/15/24
Comments:		
Tristan Cilley	<i>Tristan Cilley</i>	04/15/24
Comments:		

## Appendix B – Document Contributions

This is the current contribution of each team member towards the administrator manual.

Name	% of contribution
Allison Lupien	20% [Purpose, References, Hardware and Software Requirements, Formatting, Editing]
Jacob Michaud	20% [Installation, Known bugs, Formatting, Editing]
Maha Fazli	20% [Introduction, 3.4 user support, Formatting, Editing]
Nick Sarno	20% [Troubleshooting, Formatting, Editing]
Tristan Cilley	20% [Periodic Administration, Installation, Hardware and Software Requirements, Formatting, Editing]