VIRTUAL CLASSROOM IN VIRTUAL REALITY

Viktor Hallengren
Måns Granath

Simulering- och dataspelsutveckling , 180 Credits

Örebro, Sweden, Spring 2016

Examiner: Martin Magnusson

VIRTUELLT KLA SSRUM I VIRTUELL VERKLIGHET
Abstract
This project was created from the desire to provide a virtual training environment for teachers-in-training to practice and improve their non-verbal communication with students. The project worked by capturing the user’s movements and rendering the virtual agent’s to a screen in front of the user. Standing in front of a static screen to hold a lecture might however not feel entirely realistic. This report covers the implementation of a head-mounted display, specifically the Oculus Rift, to create a virtual reality as well as the extension of the virtual agent’s behavior and new ways to interact with the virtual agent’s. It also covers the results of an experiment where the new functionality was evaluated. The experiment was done by allowing 18 persons to test the system in both the old and new configurations and fill in questionnaires afterwards.

Sammanfattning
Preface

We would like to thank our supervisor Franziska Klügl for all the help during this project. A big thanks to all persons who went through all the tests with us as well as Emil and Sebastian for testing the Kinect and Myo.
## Contents

1 Introduction .......................................................................................................................... 6  
1.1 Background ....................................................................................................................... 6  
1.2 Project ............................................................................................................................... 6  
1.3 Objective ........................................................................................................................... 7  
1.4 Requirements ................................................................................................................... 7  
1.5 Division of Labor ............................................................................................................. 7  

2 Background .......................................................................................................................... 8  
2.1 Virtual Reality and its Uses .............................................................................................. 8  
2.2 Immersion and Presence .................................................................................................. 9  

3 Overall System Design ....................................................................................................... 10  

4 Methods and Tools ............................................................................................................. 11  
4.1 Tools ............................................................................................................................... 11  
4.1.1 Horde3D Game Engine ............................................................................................... 11  
4.1.2 Jason and the Extended Version of AgentSpeak ....................................................... 12  
4.1.3 Oculus Rift ................................................................................................................ 13  
4.1.4 Myo Gesture Control Armband ................................................................................. 13  
4.1.5 Kinect and OpenNI ................................................................................................... 14  
4.1.6 Additional Tools ....................................................................................................... 14  
4.2 Other Resources ............................................................................................................. 14  

5 Implementation ................................................................................................................... 15  
5.1 Porting the Visualisation Project ...................................................................................... 15  
5.2 Implementing the Oculus Functionality ........................................................................... 15  
5.3 Gesture Evaluation and Implementing New Ones ......................................................... 16  
5.3.1 Kinect Gestures ......................................................................................................... 16  
5.3.2 Myo Gestures ............................................................................................................ 18  
5.3.3 Gestures For Walking ............................................................................................... 19  
5.4 Implementing Additional Agent Behaviour ..................................................................... 20  
5.5 Additional Changes ....................................................................................................... 21  
5.5.1 The Classroom ......................................................................................................... 21  
5.5.2 Communication Between the Kinect Part and the Oculus Part ................................... 22  
5.5.3 Female Students ....................................................................................................... 24  

6 Result ................................................................................................................................... 25
1 Introduction

1.1 Background

The project had its background in the desire of a tool that could help teaching students train their non-verbal communication (e.g. gestures and orientation) with students. The idea was to create a 3D virtual classroom which was filled with AI-agents (virtual students) with different behavior in regard to the other agents and the teachers (the user) movements and actions. The idea to use virtual students came from the idea that the teacher would not have to feel intimidated by the the consequences that he or she made while practicing.

In an earlier iteration of this project [1] Nilsson researched different theories on human behavior, emotion and personality, and discussed about different architectures derived from these theories. The author thought that the BDI-model (Belief-Desire-Intention) suited the task the best and implemented it into the virtual students’ behavior and used in a test in the virtual environment. The BDI-architectures basic idea is that the reasoning of the agent is done in regard to what the agent believes to be true (Belief), the goal it tries to reach in the end (Desire) and the goals it committed to reach in the moment (Intentions). The students now react to the teacher’s gestures, and the other surrounding students. The teacher’s (user) gestures were captured using a motion capture system, specifically the Kinect.

The application was tested on 16 persons, 10 men and 6 women, in two different steps. First test was just to observe the students’ behavior while having a teacher that does nothing, and see if the students’ behavior seemed realistic. The second test was trying out the gestures and watching how the students would react to them. Both tests result show that the test subjects thought the realism was deemed to be of medium to high grade of realism. gestures that wasn’t captured correctly or the general lack of gestures you could make could point directly to the decrease of realism.

1.2 Project

The project consisted of several steps. The first step was to implement the VR into the virtual classroom and make so that all old features works while using the Oculus Rift, described in Section 4.1.3. For that to be possible Nilsson’s project had to be ported to a newer version of the game engine, Horde3D, described in Section 4.1.1. The second step was to further study interactions that could be implemented in the virtual classroom, such as more gestures to use for the user, the agents’ reactions to the new gestures and the use of being able to move around inside the classroom. The final step was to evaluate the finished system in an experiment to see if the new additions improved the experience.

The project started with an already implemented system for recognizing user movement with the Kinect as well as existing visualization of the classroom to the screen. There was also a system for agent behavior written in Jason.
1.3 Objective
The main motivation was to provide a virtual training environment for teachers to improve their
non-verbal communication without the use of real people or without having to consider the
consequences if they were to make a mistake. In addition, the classroom could be modified to
create different scenarios if the user wanted another challenge and each scenario is also fully
reproducible. The addition of virtual reality and extending the ways of interacting with the system
should help the user feel more presence while using the training environment which will make
the training more realistic and rewarding. This could help them to be better and more confident
teacher.

1.4 Requirements
- The user should be able to look and walk around in the virtual classroom while using the
  Oculus Rift.
- The agents should have an increased library of reactions with the teacher as he/she can
  now walk around and make gestures from different angles and distances inside the virtual
  classroom.
- Tests that make sure that the new interactions work, and that the agents responses are
  believable.
- If there were time, modify gestures and the behavior to be more accurate.
- Evaluation whether the Oculus Rift has improved presence (See Section 2.2).

1.5 Division of Labor
Everything was done together except for the Myo code which Måns did and the Kinect code
which was done by Viktor.
2 Background

Similar types of application already existed before this project. One example is TeachLivE, that was developed at the University of Central Florida. It stems from the same desire of creating a tool to help teacher train their pedagogical skills [2]. The difference was that in TeachLivE, they used professional actors to act as students, instead of AI-agents. However, the idea of still using real people to practice with, somewhat defeat the purpose of the idea to not use real students. The teacher would still feel intimidated by the actors and limit the teacher when he or she was worried about the consequences of the actions they made.

Section 2.1 goes through what VR can be used for in real life, and Section 2.2 goes over what makes a user feel presence, a feeling of being “there” in a virtual environment.

2.1 Virtual Reality and its Uses

Virtual Reality (VR) is a way of simulating a user’s presence in a virtual environment. This can be done in several ways like projecting an environment on the walls or event creating an environment with sounds in a 3D headset. The most recent development in VR is the head-mounted display (HMD). This is something that has increased in popularity in recent years with Facebook buying the Oculus, HTC releasing its own VR headset and other companies such as Sony, are working on their own VR technology. With the increase in VR technology it benefits and uses are becoming more researched.

Some areas that are currently being explored with VR are medicine and training. According to Bissonnette et al. [3] can be used to decrease anxiety when it comes to performing live music in front of an audience. In this study musicians performed a piece of music in front of a simulated audience over six sessions and their anxiety response was measured. It was shown that the anxiety to perform the same piece of music in front of a live audience was decreased between sessions and one comment from a test subject said that “... by having played a piece several times in a concert situation, I am much less nervous at the idea of replaying it in public” (Citation from [3], p 79). Other studies thought that the use of virtual environments could help user treat their glossophobia (fear of public speaking): Poeschl [4] discussed the possibility to observe the behavior of a real life audience and design a realistic virtual audience out of the data. The audience analyzed was from an undergraduate seminar and consisted of 18 men and women. The observation data was based on the different social cues in the audience, such as gestures and facial expressions. The result from the data showed that the studied audience tended to be neutral to positive.

These studies coupled with studies such as Villani et al. [5] which says you might experience more presence in VR than reality. When doing a simulated job interview they suggests that VR could be a powerful tool for learning and getting used to uncomfortable environments. There are already areas where VR is used for training, for example, the military uses it for mission training [6]. With the quick reset time and almost unlimited possibility to create different scenarios with things you cannot train on in reality makes VR a great exercise tool.
2.2 Immersion and Presence

Both these terms can seem mean the same thing when it comes to VR. “How immersed you are” or “how present you feel” in a simulation does not seem that different. But Slater [7] argues that they are two separate terms and that “Presence is a human reaction to immersion.” He also states that “Given the same immersive system, different people may experience different levels of presence, and also different immersive systems may give rise to the same level presence in different people.” So “immersion” is the feeling of being in another non-physical destination than the body is, and “presence” is the sense of believing to be in that place.

What makes a user feel a higher amount of presence is something that is being studied more and more. Wallach et al. [8] tried to see if the personality traits they described, “Empathy”, “Imagination”, “Locus of control”, “Immersive tendencies”, “Dissociation tendencies”, would affect how present the user felt during a simulation. The simulation would be tested on 84 subjects. The simulation consisted of the subjects riding an airplane and according to the theory would act differently depending on their personality traits. They found that people who are good at focusing on enjoyable activities, block out distractions as well as play a lot of games and get involved to the point of feeling like they are there, felt more presence than others [8], this personality trait is also coined as “Immersive tendencies”. There is also a study by Mikropoulos [9] which allowed children to test a game in four different configurations. The game was either projected on a wall or the user had a head-mounted display on and were either exocentric (the camera was behind the character played) or egocentric (the user was the camera). The presence reported by the test subjects was almost equal in all configurations except egocentric with a head-mounted display which was higher [9].

Slater et al. [10] wanted to test the efficiency of the virtual environments concerning the “presence response” - the perceived level of presence while interacting with the virtual environment. The goal of perceived level of presence was to be equal to a real life situation. The experiment consisted of 20 confident public speakers and 16 persons that were phobic. Half of each group would speak in front of a Virtual Environment consisted of an empty seminar room, while the other half would speak in front of a neutrally behaving crowd of five people. Conclusions shows that even with a low-level of representational and behavioral ability in the agents, the presence response was that people with phobic had no problem talking in front of the empty seminar room but would struggle in front of the five people, something that the confident speakers would not have a problem with. From this the conclusion was drawn that even with simple means, an effective virtual training environment could be produced.
3 Overall System Design

The virtual classroom is constructed as a system built up by three subsystems - a visualization system built in the game engine Horde3D, a multi-agent system built with Jason, described in 4.1.2, and a motion capture system, specifically the Microsoft Xbox 360 Kinect (Kinect). The visualization system is made up by two parts. One reads data from the Kinect about what gestures the user is doing as well as orientation, position and posture. Then it sends this information as a string over a TCP connection to the multi-agent system. The multi-agent system uses the strings it gets from the Kinect part, parses them and then uses the information to determine the agents’ different reactions. The second part of the visualization system listens for incoming strings from the multi-agent system and parses that information and either creates a new agent or updates the agent's animations and positions. It then draws this to the screen, which can be either a computer monitor or the Oculus Rift. See Figure 1 for an overview of the whole system setup.

Figure 1: An overview of the system setup.
4 Methods and Tools

An iterative approach was taken towards this project meaning that there were weekly meetings with the supervisor to discuss project advance and to plan for the following week while also adhering to the initial project specification. During the first two weeks’ information about the project was gathered. This includes what had previously been done with the project, how the agents were implemented and what theories and architecture they were using, as well as new information about immersion in a virtual reality and interaction with virtual agents. New ways of interacting with the agents was also analyzed. The following weeks were spent porting and implementing the new functionality into the project.

Too see if the new functionality increased the level of immersion an experiment was performed on the 1st and 2nd of June 2016. This evaluation was done by 18 persons, five females and 13 males who tested the system in three different ways. First by just observing the class, then interacting with it in front of a screen and lastly interacting with it with VR enabled. After each step each person filled in a questionnaire.

4.1 Tools

Since this project was a continuation of a previous project there was not much choice in what tools to use. If one library should be switched a complete rewrite of that part would be necessary.

4.1.1 Horde3D Game Engine

The Horde3D game engine is based on the open source cross-platform graphics engine Horde3D. The engine is developed at the Augsburg University in Germany. While visualization system was already built with this game engine, as seen in Figure 1, it used an outdated version which did not have support for rendering a stereoscopic camera. Thus an upgrade to the newest version was needed for this project to work.

The engine is an entity/component-based game engine with several pre-built components. This means that every object in a scene in an Entity and an entity can hold several Components. The visualization project used SceneGraphComponent for creating a scene graph to hold a list of all objects in the current scene. GameSocketComponent for creating a server or client to send and receive information. AnimationsComponent to control the agents’ animations. FACSControlComponent for controlling facial expressions. IKComponent for the agents’ bones and joints. A component written for the visualization system called ClassroomAgentComponent. The ClassroomAgentComponent, seen in Figure 1, is the component that parses the strings from the multi-agent system and updates the agents in the simulation. Every entity and component was loaded into a scene using XML-files or if needed as XML-text as a string.

Some of these components, specifically the SocketComponent and the AnimationsComponent, as well as some parts inside the core of the engine had been rewritten in the newer version and did no longer work with the visualization system.
4.1.2 Jason and the Extended Version of AgentSpeak

Jason is an interpreter for the language AgentSpeak written in Java. The programming language AgentSpeak is a logic-based language for agents using the BDI architecture. The extended version of AgentSpeak means that Jason offers user-customization using Java.

When the Java server that the multi-agent system is running receives a string from the Kinect part of the visualization system it adds these as “beliefs” to the teacher agents (The teacher is also represented by a Jason agent) by using Jasons function addBel and removes the old belief using delBel.

```java
teacher.delBel(Literal.parseLiteral("gesture(X)"));
teacher.addBel(Literal.parseLiteral("gesture(" + Gesture + ")")));```

The teacher agent is defined in an AgentSpeak file of its own. When the gesture “clap” is received from the Kinect the following code will be executed:

```agent
+gesture(clap) : true
 <=   .print("clapping, well done!");
       .send(activityMonitor, tell, teacherAction);
       ?students(Students);
       .send(Students, tell, teacherGesture(clap));
       .send(Students, untell, teacherGesture(clap)).
```

This means that if gesture clap is true everything in the block between <- and the final dot will be executed. First it will print "clapping, well done!" to the console and tell the activityMonitor agent that the teacher has done something. The activityMonitor agent just waits for a teacher action and if nothing happens it will tell the student agents that the teacher is inactive. The ? means that the agent will try to retrieve information from the its set of beliefs. This function, ?students(Students), retrieves the name of all the students. It then “tells” the students about what gesture the teacher has performed, this is the same action as addBel and delBel above but in AgentSpeak, so the gesture performed will be added to the students “belief-base”.

The agents have now been notified of the gesture that teacher has performed. The agents AgentSpeak files works the same way:

```agent
+teacherGesture(clap) : mood(best) | mood(better)
 <=   .drop_all_intentions;
       !modifyContentment(25);
       expressDisgust(1);
       lookAtCamera;
       crossArms.
```
This example is from the student “student_hostile1.asl”, the agents’ personalities is further described in 5.4. Here the agent will choose this reaction if its mood is “best” or “better”. The agent's mood is decided by an internal variable called contentment. This variable is increased by 25 in when the teacher claps in this example. The exclamation means that it’s a “goal” the agent wants to achieve, used in this case to modify its mood. The agent then looks disgusted at the camera and crosses its arms. These actions are defined in a Java class that extend Jasons class Environment. This class overrides a function called executeAction which contains definitions for all these agent actions. It looks like this:

```java
@Override
public boolean executeAction(String agentName, Structure action) {
    boolean result = false;
    ...
    else if (action.getFunctor().equals("crossArms")) {
        result = model.crossArms(agentName);
    }
    ...
    return result;
}
```

This function calls the function crossArms(agentName) which sends a string back to the visualization which updates the animation for the agent.

### 4.1.3 Oculus Rift

One of the goals of this project was to add functionality for virtual reality. This was done using a head-mounted display (HMD), specifically the Oculus Rift (Development Kit 1). The Oculus uses an HDMI connection for the screens and a USB socket.

The Oculus works by having one 7-inch screen and two lenses, one for each eye, to create stereoscopic 3D vision and has a built in head-tracking system using gyros, accelerometers and magnetometers. This head-tracking functionality was used to enable a person in the classroom simulation to look around in the room and not just be a static camera in front of the virtual agents, the students. The HMD functionality was implemented in the visualization system and to gain access to this functionality Oculus own software development kit (SDK) was used.

### 4.1.4 Myo Gesture Control Armband

The Myo armband tracks gestures and movements made with the user arms, such as making a fist, spreading the fingers or wave left or right. The Myo uses nine-axis inertial measurement units (IMU) to register the movements. The IMUs contains a three-axis gyroscope, a three-axis accelerometer and a three-axis magnetometer. The accelerometer measures the acceleration of the arm. The gyroscope registers changes in the rotation of the arm, as in yaw, pitch and roll. The magnetometer calibrates the direction of the armband. The Myo is connected wirelessly using Bluetooth, and their own SDK was used to get the data from the IMUs.
4.1.5 Kinect and OpenNI
The Kinect camera tracks the user's body using the RGB camera and depth sensor on the front of it. The depth sensor is equipped with an infrared laser and a CMOS sensor, that makes it able to capture 3D video in any light. The infrared laser is used to calculate depth in front of it. It flashes and calculates the time which the laser is reflected back and from there draws the image of the room in front of it.

The implementation of the Kinect functionality was done with OpenNI, a library for interacting with the Kinect. OpenNI allows the user to, for example, read depth data, check for users inside the camera's vision and read positions of a user’s skeleton joints. This library is very outdated and has not been updated for over three years on GitHub but unfortunately it’s a very large part of the project. OpenNI also requires the user to use 3rd party Kinect drivers and not Microsoft's own Kinect drivers.

The functionality of OpenNI is implemented in a project called KinectTracker. This project also contains all the available gestures.

4.1.6 Additional Tools
The visualization system and game engine projects was built using Microsoft Visual Studio and had to be run on a computer running Windows. Since the Oculus does not officially support the use of laptops a desktop computer had to be used. Sublime Text 2, a text editor, was used to edit the agent’s reactions and behavior. Software required to build the engine was OpenAL, a cross-platform 3D audio library (which was not used in this project but required for Horde3D to even start since it was one of its dependencies).

The visualization project was also dependent on GLFW, an open-source library for creating OpenGL contexts and receiving input. This library was used to create the window that was used for rendering. It was also used for receiving mouse and keyboard input.

4.2 Other Resources
The Kinect uses a proprietary plug so a USB adapter was needed.
5 Implementation

This chapter goes through all the updates and changes that has been made and how they were done.

5.1 Porting the Visualization Project

The project was ported by importing the projects that make up the visualization system to the new game engine project. The projects that needed to be ported was the Virtual Classroom (This was the project that ran both the visualization and the Kinect parts), the ClassroomAgentComponent and the KinectTracker project. After moving these projects to the new engine there were thousands of linker errors. This means that either some .dll, .lib source code file is missing or misplaced, or referenced code in the files does not exist. First all the .dll and .lib files were added correctly. Then the updated version of the game engine had some code that had been renamed, updated or deleted entirely so both the new engine and the project had to be updated to fit with each other. For example, to set the animation frame a class called AnimationJob was needed but in the new version it was called KeyframeAnimationJob, however the functionality was the same. In other places code had just been removed, there was for example no function to get the connection status of a socket anymore so it had to be re-implemented from the old version of the engine.

When the project was able to run there were a few bugs, the most notable one was that animations sometimes didn’t play at all. The problem was that for some reason the new game engine sometimes returned a negative value when you wanted the length of an animation so when the visualization project tried to play an animation it wouldn’t since the duration was less than zero. It was the correct duration of the animation just negative. The real solution for this was never found since everything that should have affected this looked the same as it did in the old engine. So if the length was found to be negative the minus was just removed.

5.2 Implementing the Oculus Functionality

To make the Oculus work two things had to be done. First, the rendering had to be changed to stereoscopic rendering, this functionality was new in the newest version of Horde3D. Second, the camera needed to follow the user's head movements.

As mentioned in 4.1.6 the visualization system used a library called glfw to create a window to render to however this library lacked the function to get a reference to the window which the Oculus Rift required when being configured. To get this functionality this library had to be upgraded to the newer glfw3 which meant that everything that had to do with the window and input had to be rewritten.

The rendering had to be changed to make use of the Oculus stereoscopic vision. See Figure 2 for an example of how this looks. This functionality was something that was new in the newest
version of Horde3D and needed for the Oculus to function. All required settings such as field of view and eye offset was supplied with the Oculus SDK.

The Oculus SDK supplies all the necessary information from the HMD such as position and rotation which is applied to the game camera. All the Oculus functionality was implemented in a way that would make it easy to switch between the old and new functionality. This was done using C++’s `#define` and `#ifdef` preprocessor directives. The ability to quickly switch between the old and new functionality was needed for the final test.

![Figure 2. Example of how the stereoscopic rendering looks.](image)

**5.3 Gesture Evaluation and Implementing New Ones**

The project started with some already implemented gestures. These had been created when the project first started in Augsburg, Germany. But according to Nilsson [13] some of these were just too inaccurate to be used and he had to leave out several of them leaving him with a very limited repertoire of gestures. This was something that had to be upgraded and in this chapter, ways of improving this are explored.

**5.3.1 Kinect Gestures**

The Kinect sees the user as a skeleton with different joints for head, shoulder, arms, elbows, hip, knees and feet, and would compare joints to each other to determine what gestures the user would make. In some cases, there would be several state that needed to happen in order for the correct gesture to register. All gestures mentioned in this chapter is shown in appendix A2.

The gestures used by Nilsson [13] was:
• *Silence* - Hold up either hand higher than the shoulder.
• *Clap* - Clap both hands.
• *Hold Ears* - Hold both hands over the ears.
• *Wag Finger* - Wag either hand in a right-left-right motion.

Nilsson [13] created reactions from the agents. The original system contained more gestures but had no reactions connected to them, for reasons due to low accuracy and low probability that they would register. These are:

• *Hit Table* - Move either hand in a downward motion from the shoulder at a higher speed.
• *Arms Crossed* - Hold the arms crossed.
• *Facepalm* - Hold either hand over the face.
• *Scratch Head* - Hold either hand on the side or top of the head.

To determine whether the gesture detection was accurate enough to be used in the final test of the simulation, a test was performed. Four people, with different experience with the Kinect and all were right-handed, tried each gesture 25 times. The result showed whether the gesture was *Registered*, *Not Registered* or *Several Gestures Registered*. All the possible gestures were tried using both hands, if possible. Results of this tests are shown in table 1.

<table>
<thead>
<tr>
<th></th>
<th>Hold Ears</th>
<th>Clap</th>
<th>Wag Finger(L/R)</th>
<th>Hit Table(L/R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registered</td>
<td>47%</td>
<td>70%</td>
<td>68%/64%</td>
<td>55%/72%</td>
</tr>
<tr>
<td>Not Registered</td>
<td>26%</td>
<td>21%</td>
<td>29%/32%</td>
<td>40%/27%</td>
</tr>
<tr>
<td>Several Gestures</td>
<td>27%</td>
<td>0%</td>
<td>3%/ 4%</td>
<td>5%/1%</td>
</tr>
<tr>
<td>Registered</td>
<td>53%/79%</td>
<td>67%</td>
<td>5%/2%</td>
<td>87%/86%</td>
</tr>
<tr>
<td>Not Registered</td>
<td>10%/11%</td>
<td>26%</td>
<td>24%/36%</td>
<td>9%/9%</td>
</tr>
<tr>
<td>Several Gestures</td>
<td>37%/10%</td>
<td>7%</td>
<td>71%/62%</td>
<td>4%/5%</td>
</tr>
</tbody>
</table>

*Table 1: A Table with the success rate from all the gestures implemented from the start. The percentages indicate the amount of attempts that resulted in that specific row.*

Using this data, the decision to not use the gesture *Facepalm* was made, as it was too similar to *Scratch Head* so both gestures would be registered if *Facepalm* was used. Another reason for not using the *Facepalm* gesture was because you can’t touch your face when you have the HMD on.

With the addition to move about in the room, several more gestures were implemented to accommodate the new the possibility of movement and tested during the project, which some of were used in the final tests. These were:

• *Hand on Shoulder* - Hold either hand forward as if holding someone’s shoulder.
- *Slap* - Hit with either hand as if slapping someone.
- *Explaining* - Hold either hand out, as if showing something written on the wall.

The three new gestures were also tested on the same four persons with the same test, trying the gestures 25 time each on each hand, if possible. Results of this test is shown in table 2.

<table>
<thead>
<tr>
<th>Hand On Shoulder (L/R)</th>
<th>Slap(L/R)</th>
<th>Explaining(L/R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registered</td>
<td>89%/80%</td>
<td>73%/81%</td>
</tr>
<tr>
<td>Not Registered</td>
<td>9%/10%</td>
<td>23%/16%</td>
</tr>
<tr>
<td>Several Gestures Registered</td>
<td>2%/1%</td>
<td>4%/3%</td>
</tr>
</tbody>
</table>

*Table 2: The success of the new gestures that's being implemented. The percentages indicate the amount of attempts that resulted in that specific row.*

*Explaining* was inspired and created while watching lectures on YouTube with teachers with experience in teaching. Frequent gestures used in these lectures would be added in the system. Some series watched were Thomas Pardon-McCarthy's beginners course in programming with C [14] and Harvard CS50 course [15].

*Hand on Shoulder* was suggested from the supervisor for this project, Franziska Klügl, as a way to calm down students by placing one's hand on their shoulder. As first intended this was supposed to have a counterpart student agent to place the hand on, but that counterpart is missing. *Slap* was created in need of having a gesture that is unacceptable in a classroom environment, to make students angry and surprised.

Descriptions on how to perform these gestures can be found in Appendix A2.

### 5.3.2 Myo Gestures

The Kinect could only register the bigger movement of the body, so the Myo was considered to register the smaller movements in the hands, taken from the predetermined hand gestures built-in from the Myos SDK. The Myo can be worn on either arm, and has the gestures:

- *Fist* - Clench the hand to a fist.
- *Spread Fingers* - Hold the hand straight out, and spread the fingers.
- *Wave In* - Wave the hand inwards to the body.
- *Wave Out* - Wave the hand outwards from the body.
- *Double-Tap Fingers* - Tap a finger against the thumb twice to unlock the Myo.

The Myos gestures were also tested, like the Kinect, if they were accurate enough to be used in the real simulation. This test was made by three people, all right-handed, who wore the armband on each hand and tested the gestures twenty-five times, both while standing and sitting. The result
showed whether the gesture was *Registered* or *Not Registered*. Results of this test is shown in table 3.

<table>
<thead>
<tr>
<th>Gesture Registered</th>
<th>Fist(L/R)</th>
<th>Spread Fingers(L/R)</th>
<th>Wave In(L/R)</th>
<th>Wave Out (L/R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%/96%</td>
<td>58%/74%</td>
<td>98%/97%</td>
<td></td>
<td>88%/84%</td>
</tr>
<tr>
<td>0%/4%</td>
<td>32%/25%</td>
<td>2%/3%</td>
<td>12%/16%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gesture Standing</th>
<th>Fist(L/R)</th>
<th>Spread Fingers (L/R) Standing</th>
<th>Wave In(L/R)</th>
<th>Wave Out (L/R) Standing</th>
</tr>
</thead>
<tbody>
<tr>
<td>97%/98%</td>
<td>82%/56%</td>
<td>96%/96%</td>
<td>88%/53%</td>
<td></td>
</tr>
<tr>
<td>3%/2%</td>
<td>18%/44%</td>
<td>4%/4%</td>
<td>12%/47%</td>
<td></td>
</tr>
</tbody>
</table>

*Table 3: A table showing the result from testing the Myos gestures while sitting and standing. The percentages indicate the amount of attempts that resulted in that specific row.*

However, these were the only five hand gestures the Myo was capable of recognizing. It is possible to combine movements such as making a fist and raising the arm at the same time to create new gestures. But none of these gestures seemed to make much sense in this project. The spatial data from the Myo could probably be used to detect where the hand or arm would be inside the classroom but a decision was made to skip all of the functionality the Myo had to offer.

### 5.3.3 Gestures for Walking

For the implementation of the Oculus in the virtual classroom, the need of a gesture to control the movement need to be found. Several ways of achieving this were explored.

- **Fist Gesture from the Myo Armband** - By using the Myo armband on the user’s non-dominant arm and register when the user made a fist, there could be a constant speed applied on the character to simulate a feel of walking forward.
- **Walking on the Spot Registered on the Kinect** - Having the Kinect watch the user’s legs for an alternating pattern of lifting the legs one at a time to view it as if the user was walking on the spot. Using a second Kinect just for the legs was also considered.
- **Swinging the Arms Alternating Using the Myo Armband** - By constantly swing the arms back and forth, there would be a speed applied to the character as long as the swinging continued.
- **Using a Wireless Controller or Mouse** - At an earlier phase of the project, there was an idea of using a wireless controller to control the movement, but was discarded because of the need to having the hands free to make gestures.

As stated, using a controller or mouse to move was scrapped almost immediately. The two Myo gestures, swinging the arms and making a fist were only tried with the Oculus on the head a making the gesture to see if it felt natural however neither of them did. The “fist gesture” had an
advantage in that the user could do other gestures while walking but it felt to unnatural. The Myo also had a system where it locks itself if it’s inactive for two seconds and you have to unlock it to make a gesture. This made using it even worse than it should have been. Walking on the spot with the Kinect was the only thing left but like with all things that had to do with the Kinect it was not a 100% accurate but since no other gestures involved the legs it never double registered with other gestures.

5.4 Implementing Additional Agent Behavior

The agents are Belief-Desire-Intention (BDI) based agents which means that each agent will reason about what to do given these three modules. “Beliefs” are what the agent currently believes to be true about the world. “Desires” is what it wants to achieve and “Intentions” is what it will do in the short-term to achieve those desires [11]. Each student also has a personality assigned to it inspired by the OCEAN personality model [12]. The different implemented personalities are “Extravert”, “Hostile”, “Nervous” and “Timid” and are represented by different BDI behaviors. This means that each student will have a different reaction to each gesture made by the teacher.

The virtual students have a wide range of reactions implemented. They are categorized in two different categories, “Gestures” and “Facial expressions”. In gestures there are reactions like “Wave”, “Eat” and “Point”. The facial expressions are different states of mood, which can be shown in the virtual students face in reactions to the teacher. Some examples are “Anger”, “Sadness”, “Neutrality”.

Extending the behavior of the agents was done in two steps. Firstly, the agent behavior was added by extending the already existing agent types. Secondly, since the teacher now had the ability to move around inside the classroom agents should also react to where the teacher was standing in the room.

For each new gesture the teacher agent had to tell the students about what it the user was doing:

```plaintext
+gesture(hitTable) : true
  <- .print("hitting the table");
      .send(activityMonitor, tell, teacherAction);
      ?students(Students);
      .send(Students, tell, teacherGesture(hitTable));
      .send(Students, untell, teacherGesture(hitTable)).
```

Here the teacher does the gesture “Hit Table” and tells all students about and as mentioned in 4.1.2, it will add teacherGesture(hitTable) to the students’ belief-base. Next each student has to have a reaction to each new gesture. This is an example from student_extravert1.asl:

```plaintext
+teacherGesture(hitTable) : mood(worse) | mood(worst)
  <- .drop_all_intentions;
      iactions.unaryMinus(25, Result);
      !modifyContentment(Result);
```

lookAtCamera;
expressAnger(1);
makeMockingGesture(1);
.wait(1000);
expressNeutrality(1);
expressNeutrality(1).

If the agent is in a bad mood when the teacher hits the table, the agent will look angrily at the teacher and make a mocking gesture and after a second go back to its neutral expression. Here the iactions.unaryMinus(25, Result); is also a user specified Java function for retrieving a negative value. So this agent’s mood is reduced by 25. Extending agent behavior was then done by studying the already in place behavior and trying to keep their reactions in line with their personalities. While the agent’s personalities were based on the OCEAN personality model and their architecture was based on BDI, the agent’s reactions were not grounded by scientific analysis, but were just made up by Nilsson [13]. The additional reactions were created in such a way that they would fit with the existing ones. For example, the “hostile” agent will do much more aggressive gestures when upset about something and the “timid” agent will mostly look away and look sad.

For the second part, reacting to the teacher’s position, there was already a part of the string sent from the Kinect part of the virtual classroom that contained the user’s position in front of the Kinect camera. This part was previously unused. It was changed to contain information about the teacher’s position inside the classroom. The classroom was split into nine parts, “Left”, “Center” and “Right” which each have “Front”, ”Middle” and ”Back”, so for example “centerFront” meant that the teacher was standing in front of the whiteboard which would make the students pay more attention than when the teacher was standing behind them or in a corner. The students were also made to react to the teacher entering the “centerMiddle” zone, because the students are located in that zone. Most of the students will look up at the teacher if that happens but some, for example the timid agent will look down at their papers. These reactions were also made to fit with the existing behavior of the agents.

5.5 Additional Changes

5.5.1 The Classroom

Since the user could turn around with the implementation of the Oculus and see all of the classroom some things had to be changed. Behind the camera in the old version was just an empty wall with a doorframe without a door that went out into nothing so a door was added. To increase the impression of a teaching situation a blackboard was added. See Figure 6 for a view of the new classroom.
5.5.2 Communication Between the Kinect Part and the Oculus Part

The whole system is connected with two different programs but as three separate systems, one that handles the visualization system and Oculus, one that records the user with the Kinect and last part that handles the virtual student’s behavior. But the Kinect part and the visualization run in the same program as one instance.

The computer available to run the Oculus was too old to reliably handle everything without causing the visualization to run at a very low framerate which made the visualization system unusable. So a decision was taken to run the Kinect part and multi-agent system on one computer and the Oculus on another. See Figure 7 for an overview of the setup. This setup meant that the program controlling the camera, which in the new system is the teacher’s position in the room and the program that sends information to the multi-agent system was separated. When the Kinect part was separated to another computer it could no longer check the position of the camera and informing the multi-agent system about a camera’s position in the room. The Kinect also needed to tell the visualization part on the second computer that the “walk” gesture had been performed to move the camera. To fix this both the part with the Kinect and the part with the visualization system had to run a server and client and send messages between them over a TCP-connection. It was implemented so that when activated the Oculus functionality via C++’s preprocessor directives it would act as the “Oculus server” and if you disabled the Oculus it would become the “Kinect server”. The Kinect server would then send a message every time the gesture “walk” was made and the Oculus server would send a message to the computer handling the Kinect when the teacher’s position updated. Figure 8 contains an overview of how the setup looked after this was implemented. This should not have been necessary if the computers had

Figure 6. The classroom with a door and a blackboard.
been good enough to run everything on one computer and is therefore implemented in such a way that it would be very easy to remove once it’s not needed anymore.

Figure 7. How the system worked when split between two computers.

Figure 8. After adding communication between the two parts that had been split.
5.5.3 Female Students

The supervisor for this project asked if it was possible to add more female students to the classroom since one of the complaints in the previous exam project had been that there was not an equal amount of males and females. There was already models for female students in Horde3D project but after adding them it was realized why they had been removed in the first place. The bones were not rigged correctly which made the movements, especially when it came to head movements. For example, it looked like it was going to break its neck when looking to the side. The model also had gaps between textures and the facial expressions did not look as good as the male students did. See Figure 9 for an example of all the problems with this model. Because of all these problems the decision was made to not include any female students in the classroom.

![Female model example](image)

*Figure 9. Example of how the female model behaved and looked compared to the male model.*
6 Result

6.1 The Finished System

In the end all requirements were fulfilled. The biggest difference is the addition of Oculus Rift functionality which enables the user to use a HMD too look around inside the classroom. This is described in Section 5.2 and can be seen in Figure 2. There was also TCP-communication added between the Kinect part and the visualization part allowing the user to run two instances on separate computers, this is described in 5.5.2.

The Kinect part and multi-agent system was extended to create more interactions with the agents. How these were done is described in Section 5.3 and 5.4. The biggest addition when it comes to interacting with the agents is the user's ability to walk around in the room with the HMD on. This will now make the agents react to where you are inside the classroom. See Figure 10 for an example of the user walking up close to the students.

![Figure 10. The teacher has walked up close to the students which are now looking up at the teacher. With the exception of the “timid” agent who looks down and does something else.](image)

Other than position inside the room, the new gestures available were “slap”, “explaining” and “hand on shoulder”. The reactions from the students to these can be seen in Figure 11, 12 and 13.
Figure 11. Someone got slapped. The students are surprised and angry.

Figure 12. The teacher is explaining something. Most students are writing except one who is in a bad mood.
Reactions were also added to already existing gestures that just weren’t used. These gestures were “scratch head”, “arms crossed” and “hit table”. Some new reactions to these can be seen in Figure 14, 15 and 16.
Figure 15. The teacher is standing with its arms crossed. Some students are losing interest and looks at other things.

Figure 16. The teacher has hit the table. Students are surprised or angry and a student in the back is making gestures towards the teacher.
6.2 Experiments

The experiments were made to evaluate if the implementation of the VR would increase the sense of presence, and such create a more realistic experience. The final experiment was performed on 18 different subjects, 5 females and 13 males. They would be out through three scenarios and give feedback them by answering a questionnaire. These questionnaires were created by the project supervisor, Franziska Klügl. The full version of the questionnaires can be found in Appendix A1.

- **First scenario, Observation**
In the first scenario, the subjects were told to observe a class of eight students for three minutes and evaluate how the students react to an inactive teacher, and how they interact with each other. Then they would answer the questions about how realistic they thought the students looked, behaved and behavior to the other students and then some questions about how they felt during the test. They also filled in their age, gender and experience with teaching, video games and virtual reality. The statistics about age, gender and experience can be seen in Figure 17.

- **Second scenario, Screen-based interaction**
In the second scenario, the subjects hold a lecture or pretend to, in front of the class of eight students for three minutes and try to keep the students’ attention throughout the test. They were not told which gestures the students would react to, just that the important part was their body language, not what they talk about. The test was performed with the Kinect in front of a TV-screen. After three minutes of interacting via the screen, the subjects answered the same questions as for the first scenario and some questions about how realistic they found the student’s reactions to the subjects’ action were.

- **Preparation for third scenario**
In preparation for the third scenario, the subjects were to get comfortable with wearing the HMD and trying out the walking gesture. The subjects were told to try to walk forward toward a window on the other side of the classroom and then walking back to the blackboard. When the subjects were comfortable with the movement, the third scenario were started.

- **Third scenario, VR-based interaction**
In the third and last scenario, the subjects should hold another lecture or pretend to, in front of a class of eight students for three minutes and try to keep the students’ attention throughout the test. The test will be performed in front of a Kinect while wearing the Oculus as a screen. After three minutes the subjects again would answered the exactly same questions as in the second scenario again. Finally, they answered some questions about which scenario of the two later they found more interesting and realistic. They could also add a comment if they wished to.

After all the subjects had done the experiments, the results were put together and was evaluated.
All the subjects had at least some experience in teaching, which was not planned, but was good for the experiments. If there wasn’t any teacher it could mean that they were going to more unprepared for the task, and were going to act more anxious. The ages span from the youngest at 28 and oldest at 58. The average was at 38.8. There was a lot more men involved in the experiments than women.

### 6.3 The Results of the Experiments

The difference between the two last experiments were the fact that you could walk around in the classroom and the agents somehow reacted to that. Other than that there was no difference between the two last scenarios. The students had the same behavior and the gestures stayed the same. So an important thing to look at is if the test subjects experienced the two scenarios differently.

There were three questions that stayed between all three scenarios. The first question the test subjects had to answer was “How realistic did you find the behavior of the virtual students?”. With the possible answers ranging from 1 meaning “not realistic at all”, to 5 meaning “very realistic”. The results for this question can be seen in Figure 18. The data is presented with a box-and-whisker plot. The top whisker indicates the greatest value excluding outliers and the bottom whisker the lowest value excluding outliers. The blue box is the lower quartile which means that 25% of the values are lower than the value indicated by the bottom of this box. The
gray box is the upper quartile, so the upper line of the box indicates that 25% of the values are greater than this value. The yellow dot is the median, so half of the values are greater than it and half of the values are lower.

![Figure 18. A diagram showing the results for the question “How realistic did you find the behavior of the virtual students?”](image)

While the VR-ratings are not significantly higher than the screen-based ratings the median is back at three as in the observation scenario. Reasons for this might be that the user feels more “observed” by the students with the HMD on since the user becomes the camera and is not just looked at through a screen. It might also just be because of the resolution of the HMD which could make it harder to see exactly what the students are doing. Facial expressions are very hard to see with the HMD on and because of that the user has less ways of interpreting what the students are doing. In Nilssons [13] experiment the test subjects also rated the realism of the agent’s behavior lower when interacting with the students than when just observing. The fact that it gets rated higher in the VR-based scenario again could indicate that it adds a bit of presence.

The second question was “How realistic did you find how the virtual students interacted with each other?”. This question uses the same scale as the previous one. The results for this question is shown in Figure 19.
The results for this question is also not significantly different in any of the scenarios to be conclusive, but as with the previous question, slightly higher for the VR-based scenario. Again, this might have to do with the difficulty of seeing exactly what the students are doing, and what their facial expression is with the HMD.

The third question was “How realistic did you find how the students looked like?”. The results for this question can be found in Figure 20.
Figure 20. A diagram showing the results for the question “How realistic did you find how the students looked like?”.

These results show even more what was mentioned in the two previous questions about how hard it is to see the students in the HMD because of the low resolution. It could also be that given the ability to move closer to the students and inspect them more closely made the models appear worse than when viewed from a distance.

Three questions were added to compare the screen-based interaction and the VR-based interaction. The first question was “Do you think you had full control over what you did?”. The result can be seen in Figure 21.
Figure 21. A diagram showing the results for the question “Do you think you had full control over what you did?”. The value of the blue bar represents the average value from the questionnaires.

This shows that the user felt a lot less in control in the VR-based scenario. A reason for this could be the walking. While every person was allowed to take some time to learn walking inside the classroom most people still had problems using it. The inaccuracy of the Kinect also created a problem when it sometimes registered a gesture that wasn’t done. This usually was not a problem and went unnoticed but when it registered a walk gesture and the camera started moving forward, without the user actually intending to move, it’s easy to understand why the user would feel less in control of what’s happening.

The next question was “How realistic did you find how the students interacted with you as teacher?”. See Figure 22 for the results.
The VR-based scenario is ranked slightly higher than the screen-based scenario. The fact that there’s no difference in how the students interact with the user seems to indicate that being the camera and not look at the students through a screen adds a bit of presence to the scenario.

The final question was “How well did the virtual students react to your activities as teacher?”. The results for this question can be seen in Figure 23.
There is really no difference in how the users felt that the students reacted to their actions between the two scenarios. This should be the expected result as the students are programmed the same way in both scenarios with the exception being that agents react to the user's position in the VR-based scenario. Even though the median is very average the standard deviation is very high so the test subjects had a very varying opinion on how much the students reacted to them. This could be because everyone has different expectations of how a class should act given the situation they acted out, the agents' behavior in the observation only scenario and their prior teaching experiences.

The final task for each scenario was to fill in a series of scales between two opposite emotions about how they felt during their interaction with the students. This was done to see if the users felt any different between the three scenarios, if so it could be an indication of feeling more presence. These emotion-pairs were for example nervous/relaxed, bored/interested and powerless/powerful. See Appendix A1 for a list of all emotion-pairs. The result of these emotion-pairs are presented in Figure 24. If the pair is nervous/relaxed, 1 means nervous and 5 is relaxed.

![Figure 24](image)

*Figure 24. A diagram showing the results from the question “How did you feel while observing the virtual students?” A lower value indicates the first in a pair so dizzy/clear a 1 is dizzy and 5 it clear.*

While most emotion-pairs favored the VR-based configuration by a small amount they don’t show anything significant or especially interesting. All average values except two were within 0.3 points of each other when comparing the screen-based and VR-based scenarios. The only pair that did show something significant was dizzy/clear. See Figure 25 to see a more detailed diagram of this result.
Figure 25. A diagram showing the result of the scale dizzy(1)/clear(5).

This was something that many test subjects left comments on as well. One person commented “The VR scenario can really make someone sick (me...)”. Another commented “Turning around 180 & back makes you very dizzy. That should be a quite natural behavior during lessons.”. This phenomenon is called “simulator sickness” and is slightly different from regular motion sickness. Instead of being induced by actual motion it comes from when a simulated environment signals self-motion but there’s a lack of any actual motion [16]. Oculus Developer Center mentions some reasons for this and the most relevant ones for this project is flickering image, input latency and low framerate. Using a newer version of the Oculus or a more powerful computer would probably help reduce this discomfort.

The final page contained three questions regarding the two last scenarios. The results of these questions can be seen in Figure 26. In this diagram a value of 5 represents the VR-based scenario and 1 the screen-based scenario.
The first question “Which scenario did you find more interesting?” shows that people clearly favored the VR-based scenario. The last two questions “In which scenario was the behavior of the students more realistic?” and “In which scenario did you feel able to influence how the virtual students reacted?” went very slightly in favor of the VR-based scenario.

While there was nothing statistically conclusive from the questionnaires it still shows a slight favor towards VR. Even with all the problems the VR-based scenario had with dizziness and problems moving around people still preferred it to the screen-based configuration even if just slightly. One person commented “But with VR it felt like I “was there”.”. These results suggest that VR is the way to go with this project.
7 Discussion

7.1 Limitations

There are several limitations to this project. The biggest was the use of old libraries. Some of them were upgraded during this project like glfw to glfw3 and the new version of the Horde3D engine. But others are impossible to upgrade without a complete rewrite. The library used to get data from the Kinect, OpenNI, was abandoned several years ago and it was hard to even find drivers and installers that worked together to get the visualization system working. The drivers for the Kinect camera were so old that they didn’t have a digital signature which is required to install drivers on Windows 7 and newer unless you turn driver signatures off manually when booting the computer.

Even though the game engine was upgraded the graphics still feels out of date, this might be because of the old character and environment models and it could probably look better with new models but things such as flickering shadows and other graphical artifacts (Some can be seen on the wall of Figure 9.) suggests that the engine is getting too old. Another problem with OpenNI and Horde3D is the almost complete lack of documentation and community around them. If you encounter a problem with anything it’s completely up to yourself to solve it.

A problem that Nilsson [13] also pointed out is with the motion recognition system. It can only really recognize large movements and even then it’s still really inaccurate. Since it has a hard time recognizing even large movements the number of gestures you can make becomes really limited. Since small movements are pretty much undetectable it’s hard to create gestures that are generally done, for example, holding both ears are not something that a teacher would often do. Since small details is a large part of body language it’s a very limiting factor for how realistic this project can become.

Jason also seems to be a limiting factor in this project because of how resource intensive it is. Running a Jason on the same computer as the visualization system dropped the framerate to unusable levels which meant that it had to be run on a separate computer. Running an AI system should not require a separate computer.

7.2 Compliance with the project requirements

As stated in chapter 1.4 the requirements were:

- The user should be able to look and walk around in the virtual classroom while using the Oculus Rift.
- The agents should have an increased library of reactions with the teacher as he/she can now walk around and make gestures from different angles and distances inside the virtual classroom.
- Tests that make sure that the new interactions work, and that the agents responses are believable.
● If there were time, modify gestures and the behavior to be more accurate.
● Evaluation whether the Oculus Rift has improved presence (See Section 2.2).

The Oculus functionality was implemented and the teacher has the ability to walk and look around the room using the Oculus HMD.
The agents also react to the teacher’s position in the room, described in 5.4.
Tests with the Oculus and without was done on the 1st and 2nd of June 2016 and the results were evaluated afterwards.

Additional gestures were implemented. Those and the old ones were evaluated on how often the gesture was recognized, described in 5.3.1. The new gestures were also tested to see if the agents reacted to them.

Some extra work was added towards the end of this project such as changing the classroom environment, described in 5.5.1 and adding TCP-communication between the Kinect and Oculus parts when running on two different computers, described in 5.5.2.

A few things should probably have been done in another way. The things that went wrong were mostly down to bad preparation before the project started. The project should have been ported to the new engine before the project started because the visualization system project was not even runnable when the project started absorbing a lot of time to just make it work. Then a lot of time were spent with bad hardware. Computers that could not run the Oculus, problems with Kinect drivers and lastly hardware that could not run the entire project slowed down progress.

7.3 Special Results and Conclusion

While nothing conclusive came out of the tests they show a small increase in how interested and engaged the person is when interacting with the students with VR. This shows that VR is the way to go forward with this project but that there’s a lot of room for improvement (See Section 7.4).

7.4 Future Project Development

The training environment has been met with problems along the way of its construction. A lot of the problems have been the result of the lack proper hardware, or could be improved with an upgrade in hardware. For future development, these things would like to be addressed. For example:

● Replacing the Oculus Rift with another head-mounted VR-display, like the HTC Vive, which is made as a room-scaled experience which could be used to walk around without using gestures in front of the Kinect.
● The Vive also has two controllers to detect where your hands are in the space around you which could be used to simulate arms and/or hands in the simulation. This could also be a use for the Myo armbands.
- Adding a microphone so that voice commands can be implemented and further make the experience more realistic.
- Having something checking the user’s posture and make the agent’s react differently depending how the teacher is standing.
- To be able to register the user’s facial expressions could be a useful feature however it would be difficult to register while wearing a HMD.
- A larger project would be to transfer the project from the Horde3D game engine to a simpler and more supported game engine, for example Unity.
- Updating and creating new models for the students and environment to look more realistic.
- By adding more reactions to the virtual students it could make them more diverse and realistic.
- Extend agent behavior to allow the user to communicate with individual students and not just the entire class at once.
- Creating better and more fluid animations would make the agent’s more realistic.
- Having a more powerful computer would remove the need for two separate computers and make the system easier to setup and use.

7.5 Reflection on own learning

7.5.1 Knowledge and comprehension
K1 - During this course a lot of time and energy has been gone into understanding the old system and libraries and also getting them to work. A lot of experience has been gained in troubleshooting hardware and software, often without any documentation or community to get help from.
K2 - Working with the Oculus Rift greatly increased the interest and knowledge in VR technology. We have started to think about what other applications or games VR could be used for in the future.
K3 - Reading about presence and immersion has also increased our knowledge about what makes for a good VR experience.
K4 - The experiments were also a real learning experience. None of us had conducted an experiment with real persons before. There was also a lot of data to analyze which also was a great learning experience.

7.5.2 Proficiency and ability
P1 - The goal for this project was defined as several parts in chapter 1. Firstly, the old system was going to have VR implemented and make that work, secondly the library of the virtual students was to be extended. Finally, the system was going to be evaluated to see if the first two step was successful to make the system more realistic and thus more usable.
P2 - The references were mostly scientific papers, books and articles. Most were found by using Summon at oru.se, but some were found as references in other reports or on the internet.
P3 - A weekly plan was put together for the 10 weeks the course was held. The plan involved time for research, learning about the old system and methods and implementing the planned
changes. The plan was mostly followed accordingly with some minor changes. Amongst these changes were the choice to push forth the experiments to after the final oral presentation because of some fixes we wanted to make.

P4 - An experiment has been done, the results are analyzed and discussed in chapter 6.
P5 - We have tried writing the report in an understandable way, yet without trying to over explain anything.
P6 - We think we succeeded in presenting our work in an interesting way during the oral exam, even though it wasn’t very long. The illustrations used in the presentation is also believed to be sufficient for explaining the setup.

7.5.3 Values and attitude

V1 - As mentioned in chapter 1.2, the main usage of this final system will be as a training environment for teachers to improve their non-verbal communication skills.
V2 - Within this report, most of the research have been about VR, a field that is growing very quickly with the recent releases of the Oculus Rift and similar HMDs. We have in chapter 7 discussed the possibility of using other HMDs such as the HTC Vive or newer versions of the Oculus Rift.
V3 - We have had meetings with our supervisor once a week to keep her updated on what we have been doing. She also helped us find subjects for our experiments, which we think we handled professionally.
V4 - We have documented that which we think is required in this report, as well as in the appendices, to be understood by someone that is not that experienced in this field.
8 Acknowledgements

Thanks to all the persons who took part in the evaluation of the project, Franziska Klügl and the persons developing this project before.
9 References


[15] Harvard, CS50, Computer Science[Internet], https://www.youtube.com/user/cs50tv

Appendices

A1

Observation Questionnaire

<table>
<thead>
<tr>
<th>Gender:</th>
<th>Male</th>
<th>Female</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Prior Experience with Teaching: Professional
Prior Experience with Video Games: Expert
Prior Experience with Virtual Reality: Expert

How realistic did you find the behavior of the virtual students?

Very realistic  o  o  o  o  o  o  not realistic at all

How realistic did you find how the virtual students interacted with each other?

Very realistic  o  o  o  o  o  o  not realistic at all

How realistic did you find how the students looked like?

Very realistic  o  o  o  o  o  o  not realistic at all

How did you feel while observing the virtual students?

dizzy  o  o  o  o  o  o  clear
nervous  o  o  o  o  o  o  relaxed
upset  o  o  o  o  o  o  content
depressed  o  o  o  o  o  o  excited
tired  o  o  o  o  o  o  alert
bored  o  o  o  o  o  o  interested
powerless  o  o  o  o  o  o  powerful
Do you think you had full control over what you did?
Yes, full control o o o o o o No, no control

How realistic did you find the behavior of the virtual students?
Very realistic o o o o o o Not realistic at all

How realistic did you find how the students interacted with you as teacher?
Very realistic o o o o o o Not realistic at all

How well did the virtual students react to your activities as teacher?
Intensive reactions o o o o o o Students did not react

How realistic did you find how the virtual students interacted with each other?
Very realistic o o o o o o Not realistic at all

How realistic did you find how the virtual students looked like?
Very realistic o o o o o o Not realistic at all

How did you feel while observing the virtual students?
dizzy o o o o o o clear
nervous o o o o o o relaxed
upset o o o o o o content
depressed o o o o o o excited
tired o o o o o o alert
bored o o o o o o interested
powerless o o o o o o powerful
confused o o o o o o confident
uncomfortable o o o o o o amused
**Questionnaire after VR-based interaction**

Do you think you had full control over what you did?
Yes, full control o o o o o o
No, no control

How realistic did you find the behavior of the virtual students?
Very realistic o o o o o o
Not realistic at all

How realistic did you find how the students interacted with you as teacher?
Very realistic o o o o o o
Not realistic at all

How well did the virtual students react to your activities as teacher?
Intensive reactions o o o o o o
Students did not react

How realistic did you find how the virtual students interacted with each other?
Very realistic o o o o o o
Not realistic at all

How realistic did you find how the virtual students looked like?
Very realistic o o o o o o
Not realistic at all

How did you feel while observing the virtual students?
dizzy o o o o o o clear
nervous o o o o o o relaxed
upset o o o o o o content
depressed o o o o o o excited
tired o o o o o o alert
bored o o o o o o interested
powerless o o o o o o powerful
confused o o o o o o confident
uncomfortable o o o o o o amused

**Questionnaire after both types of interaction**
Now, think about both interactive scenarios - the front situation (Screen) and immersive situation (VR Glasses):

Which scenario did you find more **interesting**?

Front Scenario  o  o  o  o  o  o  Immersive Scenario

In which scenario, the behavior of the virtual students was more realistic?

Front scenario  o  o  o  o  o  o  Immersive Scenario

In which scenario you felt more being **able to influence** how the virtual students react?

Front Scenario  o  o  o  o  o  o  Immersive Scenario

**Do you have any comments you want to share with us?**
A2

The Gestures available

Arms Crossed
Clap

Explaining - Can be done with either hand

Facepalm - Can be done with either hand
Hand on shoulder - Can be done with either hand

Hit table - Can be one with either hand
Hold ears

Scratch head - Can be done with either hand
Silence - Can be done with either hand

Slap - Can be done with either hand
Wag finger - Can be done with either hand

Walking - Can be done with either leg